UNIVERSITY OF TWENTE.

THE UNIVERSITY of ADELAIDE

Master Mechanical Engineering

Mechanics of solids, surfaces & systems (MS^3)

Multiple framework modelling and controller design of 4 link snake robot

Internship

4 months (20 ECTS)

Supervisor University of Twente:	Prof.Dr.Ir. *A. de Boer*
Supervisor University of Adelaide:	Assoc. Prof. *R. Akmeliawati*
Student:	*M.P.A. van Bergen*, s1460854
Contact:	m.p.a.vanbergen@student.utwente.nl

Adelaide, September 12th, 2019

Contents

1 Summary **iv**

2 Introduction **1**

3 Problem definition **2**

4 Analysis **3**
4.1 Snake locomotion . . . 3
4.2 Kinematic analysis . . . 4
4.3 Friction model . . . 4
4.4 Dynamic analysis . . . 5
4.5 Control . . . 8

5 Method **12**
5.1 Method step 1 . . . 13
5.2 Method step 2 . . . 13
5.3 Method step 3 . . . 14
5.4 Method step 4 . . . 15

6 Results and Discussion **16**
6.1 Results and Discussion step 1 . . . 16
6.2 Results and Discussion step 2 . . . 19
6.3 Results and Discussion step 3 . . . 19
6.4 Results and Discussion step 4 . . . 19

7 Conclusion **22**

8 Recommendations **22**

A Appendix **25**
A.1 Calculation matrices for modelling . . . 25
A.2 Serpenoid curve . . . 26
A.3 Viscous Friction model . . . 26
A.4 Coulomb Friction Model . . . 27
A.5 Visualization of propulsive forces . . . 27
A.6 Newton-Euler modelling: symbol definitions . . . 28
A.7 Lagrangian transformation to generalized coordinates . . . 28
A.8 port-Hamiltonian Input-State-Output representation snake robot . . . 28
A.9 Newton-Euler controller design: symbol definitions . . . 29
A.10 Newton-Euler controller design: controller equations . . . 29
A.11 port-Hamiltonian Input-State-Output coordinate transformation . . . 29
A.12 $\frac{dC_i}{dt} = 0$ for the second and third Casimir . . . 30
A.13 Energy-Casimir method: control laws . . . 30

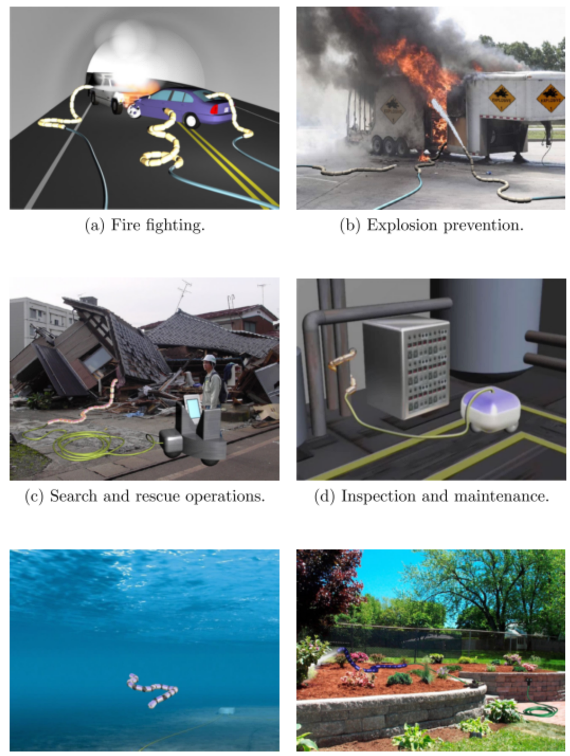
B Reflection report **33**

1 Summary

This research focuses on the modelling and controller design of snake robots. Snake robots are hyper-redundant robotic systems. This property offers unique advantages in application. Their high (internal) mobility gives them unique terrain adaption possibilities. Snake robots may be applied in search and rescue missions in disaster area's or (nuclear) area's unsafe for human inspection. However, the hyper-redundancy also causes snake robots to be very complex, highly non-linear, underactuated mechanical structures that can sometimes also be non-fully observable. A snake (robot) propels it self forward by ground friction acting on its body. All these properties pose significant challenges in modelling and controller design for snake robots. In this research the kinematics and dynamics of a 4-link snake robot moving over a flat surface are modelled in 3 different frameworks. Its Newton-Euler, Euler-Lagrange and port-Hamiltonian equations of motion are derived and simulated in MATLAB and Simulink. These models and the resulting simulations are verified and turned out to be correct. Based on the derived port-Hamiltonian model of the snake robot a port-Hamiltonian shape controller is derived with the passivity based Energy-Casimir method. Numerous possible Casimirs and controller parameters are tested. The performance of the port-Hamiltonian shape controller is compared to the performance of an established simple joint controller (derived in the Newton-Euler framework). The performance of the port-Hamiltonian shape controller turns out to be poor. The control is unstable and has poor tracking performance. Based on the possible reasons of poor performance different recommendations for the future are given related to the port-Hamiltonian controller design.

2 Introduction

One of the first snake robots is developed by S.Hirose in 1972 [8]. From this moment on snake robots are a (new) field of research. Snake robots are robots consisting of serial connected link and joint modules where the joints can bend about one or multiple axis. Snake robots are called hyper-redundant robotic systems because they have a very large number of degrees of freedom. The number of degrees of freedom is (way) higher than its desired motion degrees of freedom. This property makes snake robots an interesting class of mechanisms for research. Hyper-redundant systems pose extra modeling and control freedom but also extra challenges. To move forward a snake robot requires a synchronized motion of its whole body. Due to the many degrees of freedom the snake robot dynamics are very complex and non-linear. Furthermore the physical design of the robot requires a complex set of sensors and actuators. On the other hand however they also offer unique advantages in applications. Snake robots can move efficiently through rough and uneven terrain. Their high (internal) mobility gives them unique terrain adaption possibilities. Snake robots may be applied in search and rescue missions in disaster area's or (nuclear) area's unsafe for human inspection [11],[16],[18]. Furthermore they could also be applied in firefighting applications as hydraulic snake robot [11]. Because snake robots can move in a very versatile environment they can also be designed as amphibious robots [12]. A schematic overview of possible applications of snake robots is given in figure 2.1.



(a) Fire fighting. (b) Explosion prevention.

(c) Search and rescue operations. (d) Inspection and maintenance.

(e) Subsea operations. (f) Domestic applications.

Figure 2.1: Schematic representation of possible applications of snake robots [11]

This research focuses on flat surface (2D) snake locomotion without sideslip constraints. This assumption makes the modelling of the snakes shape and resulting locomotion more realistic than modelling with the sideslip constraint (a more elaborate description of the sideslip constraint is given in section 4.2).

There are already various models of 2D snake robot locomotion without the sideslip constraints. The Newton-Euler formulation of the equation of motion on a flat surface is developed by [12] and extended to locomotion on a slope in [13]. The same equation of motion but obtained from first-principles is derived by [22]. More models of 2D snake robot locomotion without sideslip constraint are presented in [4], [6] and [10].

Ground friction allows a snake (robot) to move in a certain direction. This ground friction can be modelled as coulomb or as viscous friction. Models with isotropic ground friction are presented by [1] and [20]. [9] studied the effect of anistropic ground friction on the locomotion of a snake.

Because of the unique features of snake robots the possible applications are very promising. This internship research contributes to the further understanding and development of snake robots by extending the modelling and controller design options applicable to snake robots.

The research presented in this report is part of a bigger project. The goal of this project is to develop a snake robot that can be used in search and rescue missions. The mechanical design and production of the snake is done by others within the project.

To keep the report concise it is unfortunately not possible to discuss everything done in the research in detail. For more detailed analysis and derivations there are various references to literature in the report and more derivations and formula's are included in the appendices as well.

3 Problem definition

In this section the general problem definition and research goals are described. Common definitions in the field of snake robot modelling and control are used in this section. More in dept analysis of the snake robot locomotion, modelling, control and related definitions and terms can be found in chapter 4.

Among others Pål Liljebäck [11] derived the kinematics of a N link 2D snake robot. He also derived the Newton-Euler equations of motion. To control a snake robot's joints he proposed partial feedback linearization in combination with a simple joint controller (or exponentially stable joint controller). With these controllers he wants to control a snake robot's shape for locomotion via lateral undulation. The first goal of this research is to simulate Liljebäck's proposed model and to control it with the proposed controller to verify the correctness of the model and to get a better understanding of snake robot dynamics.

The same snake robot motion described by the Newton-Euler equations of motion can also be derived via the Euler-Lagrange equations of motion. This is already done by [21] but with the sideslip constraint. Therefore the second research goals is to derive these Euler-Lagrange equations of motion for a snake robot moving over a flat surface without the sideslip constraint. The resulting equations of motion should give the same motion of the snake as the Newton-Euler model from [11].

A third way of modelling dynamical systems is via port-Hamiltonian modelling. Modelling systems as port-Hamiltonian systems offers advantages over other modelling methods. The method is based on the exchange of power and energy between different (multi-physical) system (parts). Port-Hamiltonian modelling should make the physical interpretation of complex (multi-physical) systems easier. Port-Hamiltonian modelling is really modelling for control. To our best of knowledge nobody has yet modelled a snake robot as a port-Hamiltonian system. Therefore the third research goal is to determine and verify the equations of motion of the snake robot via port-Hamiltonian modelling. After that the advantages of port-Hamiltonian modelling are exploited to come up with a port-Hamiltonian controller to control the snake robot's shape. The design of this controller is the fourth research goal.

Summarizing the research goals are:

1. Simulate and control the Newton-Euler equations of motion as proposed by Pål Liljebäck [11]
2. Derive, control and simulate the Euler-Lagrange equations of motion
3. Derive, control and simulate the port-Hamiltonian equations of motion
4. Design a snake robot shape controller in the port-Hamiltonian framework

Symbol	Description	Vector form
x_i, y_i	x and y postion of center of gravity (CoG) of link i	$\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^N$
θ_i	(Absolute) angle of link i with the positive x-axis	$\boldsymbol{\theta} \in \mathbb{R}^N$
p_x, p_y	x and y position of the CoG of the snake	$\boldsymbol{p} \in \mathbb{R}^2$
ϕ_i	The relative angle between link i and link $i+1$	$\boldsymbol{\phi} \in \mathbb{R}^{N-1}$
l	Half length of the links of the snake robot	
m	Mass of the link of the snake robot	
N	Number of links of the snake robot	
J	Rotational inertia of the links of the snake robot	
$f_{R,x,i}$, $f_{R,y,i}$	Friction force acting on the CoG of link i in the x or y direction.	$\boldsymbol{f_{R,x}}, \boldsymbol{f_{R,y}} \in \mathbb{R}^N$
u_i	Applied torque by actuator i	$\boldsymbol{u} \in \mathbb{R}^{N-1}$
$h_{x,i}$, $h_{y,i}$	Joint constraint force (in x or y direction) from link $i+1$ on link i	$\boldsymbol{h}_x, \boldsymbol{h}_y \in \mathbb{R}^{N-1}$
$h_{x,i-1}$, $h_{y,i-1}$	Joint constraint force (in x or y direction) from link $i-1$ on link i	$\boldsymbol{h}_x, \boldsymbol{h}_y \in \mathbb{R}^{N-1}$

Table 4.1: Symbols used to describe the kinematics and dynamics of the snake robot together with a description and their vector form.

4 Analysis

This section goes in more detail about snake (robot) locomotion, snake robot kinematics, snake robot dynamics and the different mathematical steps to derive the different equations of motion for a snake robot. Furthermore two controller design methods are discussed. The symbols in this report used to describe the snake robot are defined in table 4.1. Further more throughout this report matrices and vectors are indicated with **thick** symbols. Also throughout this report it is assumed that the mass, length and shape of every link of the snake robot is the same. This results in m, l and J being scalars (without subscript i).

4.1 Snake locomotion

Snake robots are inspired by real snakes. Real snakes generate locomotion in four different ways. They do this by lateral undulation, concertina locomotion, rectilinear crawling and sidewinding (or crotaline) [5],[8],[11] and [15]. There are also snakes that can climb and jump off tree branches. The different mechanisms for locomotion of snakes are applied in different situations and work on different physical principles. Therefore the mechanisms are also modelled differently. This research focuses on snake robot locomotion obtained via lateral undulation.

Lateral undulation (also called serpentine locomotion) is the most common and therefore also most researched mechanism for the locomotion of snakes. For this locomotion the snake sends a serpenoid curve down its body [5], [7], [11] and [22]. In figure 4.1 this is shown schematically. For more information on the serpenoid curve see appendix A.2. The propelling forces are generated by the friction between the snake's body and the ground. For the locomotion the snake makes use of anisotropic ground friction. Anisotropic friction conditions are a necessary requirement for the snake to move forward. Mathematical models for describing the snake dynamics are presented in [4], [11], [12], [14], [19] and [22]. Lateral undulation is mainly applied by snakes in open rough terrain with not too many obstacles.

Figure 4.1: Schematic representation of lateral undulation of a snake [10]

Besides the 4 types of locomotion that snakes use in nature there are also various snake robot locomotion mechanisms developed that are different from those applied by snakes in nature. Examples are snake robots with active wheels [3]. Other mechanisms like lift rolling and lean sinus-lifting are described in [8].

4.2 Kinematic analysis

To set up the dynamic model for lateral undulation of snake robots the approach from [11] is used. This approach is common since [17] and [22] also used it. For this approach specific matrices and vectors are used for the calculations. The definitions of these vectors and matrices can be found in appendix A.1. A concise derivation is given here. For the elaborate derivations see the thesis of Pål Liljebäck [11]. Within snake robot modelling the different models distinguish from each other based on what they model (2D/3D motion, type of locomotion mechanism, etc) and assumptions they make (on type of friction and certain constraints). The sideslip constraint assumes that the snake's body cannot move sideways. It is however more realistic (and complex) to model the snake robot without the sideslip constraint. This posses other requirements on the friction (model) to make the snake move forward. If the sideslip constraint is not applied the snake needs anistropic friction forces to move forward [11]. This research uses models without the sideslip constraint.

The modelling starts with the kinematic analysis. The snake robot is modelled as an open kinematic chain with N links. The kinematic parameters of the snake robot are shown in figure 4.2.

The center of mass position and orientation coordinates of link i (x_i,y_i,θ_i) are defined in the global coordinate frame. The vector's of all 1 till N x_i, y_i and θ_i positions are denoted $\mathbf{X}$, $\mathbf{Y}$ and $\boldsymbol{\theta}$ respectively (see table 4.1). Assuming that all links have the same mass m and length $2l$, the position of the center of mass of the snake robot is defined as in equation 4.1:

$$\boldsymbol{p} = \begin{bmatrix} p_x \\ p_y \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \mathbf{e}^T\mathbf{X} \\ \mathbf{e}^T\mathbf{Y} \end{bmatrix} \tag{4.1}$$

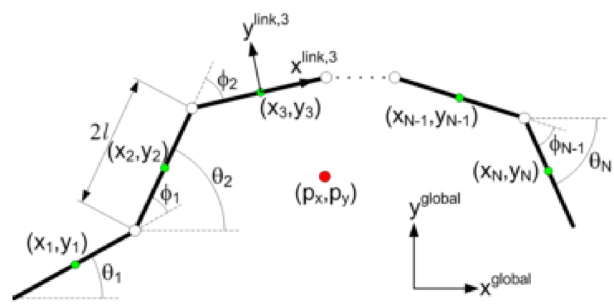


Figure 4.2: Kinematic parameters of the snake robot [11]

The set of generalized coordinates for the snake robot is $\boldsymbol{q} = \begin{bmatrix}\boldsymbol{\theta} & \boldsymbol{p}\end{bmatrix}^T$. Via the kinematic constraints at the joints the relation between the link coordinates $\mathbf{X}$,$\mathbf{Y}$,$\boldsymbol{\theta}$ and the generalized coordinates $\boldsymbol{q}$ can be derived and are given in equation 4.2.

$$\mathbf{X} = -l\mathbf{K}^T cos\boldsymbol{\theta} + \boldsymbol{e}p_x \tag{4.2a}$$
$$\mathbf{Y} = -l\mathbf{K}^T sin\boldsymbol{\theta} + \boldsymbol{e}p_y \tag{4.2b}$$

Where $\mathbf{K} = \mathbf{A}^T(\mathbf{D}\mathbf{D}^T)^{-1}\mathbf{D} \in \mathbb{R}^{N\times N}$ (see appendix A.1 for definitions of $\boldsymbol{A}$ and $\boldsymbol{D}$).

Now all the positions of the snake links (centers of mass) are expressed in the generalized coordinates. From differentiating 4.2 the linear velocities of the links can be calculated from $\boldsymbol{q}$ and $\dot{\boldsymbol{q}}$ with equation 4.3.

$$\dot{x}_i = \dot{p}_x - \sigma_i \boldsymbol{S_\theta}\dot{\boldsymbol{\theta}} \tag{4.3a}$$
$$\dot{y}_i = \dot{p}_y - \sigma_i \boldsymbol{C_\theta}\dot{\boldsymbol{\theta}} \tag{4.3b}$$

The definition of σ_i, $\boldsymbol{S_\theta}$ and $\boldsymbol{C_\theta}$ are in appendix A.1.

4.3 Friction model

The forces that propel the snake robot forward are friction forces. To model these friction forces different models can be used. Among others there is the coulomb friction model and the viscous friction model. The viscous friction model is a bit simpler than the coulomb friction model which makes the viscous friction model more suitable for analysis and control purposes. Therefore in this research the viscous friction model is used. For both models it is a necessary condition that the friction is anisotropic ground friction (as also mention in section 4.1).

In both models $\boldsymbol{f}_R = \begin{bmatrix}\boldsymbol{f}_{R,x} & \boldsymbol{f}_{R,y}\end{bmatrix}^T$ is the vector with the friction forces acting on the links. $\boldsymbol{f}_{R,x}$ and $\boldsymbol{f}_{R,y}$ are vectors with the friction forces in the link's centers of mass in x and respectively y direction in the global coordinate frame. This results in $\boldsymbol{f}_R \in \mathbb{R}^{2N}$. The viscous friction model is briefly discussed in section 4.3.1. The Coulomb friction model in appendix A.4.

4.3.1 Viscous friction

The model for the viscous friction forces on the links is given in equation 4.4.

$$\boldsymbol{f}_R = -\begin{bmatrix} c_t(\boldsymbol{C_\theta})^2 + c_n(\boldsymbol{S_\theta})^2 & (c_t - c_n)\boldsymbol{S_\theta}\boldsymbol{C_\theta} \\ (c_t - c_n)\boldsymbol{S_\theta}\boldsymbol{C_\theta} & c_t(\boldsymbol{S_\theta})^2 + c_n(\boldsymbol{C_\theta})^2 \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}} \\ \dot{\boldsymbol{Y}} \end{bmatrix} \tag{4.4}$$

c_t and c_n are the viscous friction coefficients in respectively the tangential and normal direction of a link. To derive equation 4.4 the rotation matrix $\boldsymbol{R}^{global}_{link,i}$ (see appendix A.1) and its transpose are used to transform local friction forces to friction forces in the global coordinate frame. A more detailed derivation is given in appendix A.3.

From equation 4.4 follows (as expected from physics) that the friction force is opposite to the direction of motion of the links.

4.3.2 Propulsion force

With the viscous friction model and the states of a snake robot at a certain moment in time the propulsive force can be calculated via equation 4.5.

$$F_{prop} = -F_x(\theta_i)\dot{x}_i - F_y(\theta_i)\dot{y}_i = -\sum_{i=1}^{N}((c_t cos^2(\theta_i) + c_n sin^2(\theta_i))\dot{x}_i + (c_t - c_n)sin(\theta_i)cos(\theta_i)\dot{y}_i) \tag{4.5}$$

When studying these equations a bit more in depth it becomes clear that $c_n > c_t$ is required to make a snake robot move in the forward direction. Furthermore the product $F_x(\theta_i)\dot{x}_i$ is positive when the snake is moving in forward direction. Therefore this term is not contribution (actually opposing) to forward propulsion of the snake. The propulsive force $|F_{prop,i}|$ can be increased by increasing the sideways link velocity $|\dot{y}_i|$, increasing c_n relative to c_t and by increasing $|\theta_i|$ up to a maximum of $|\theta_i| = 45°$. A link i produces the most propulsive force when $\theta_i = \pm 45°$. A visual impression of the propulsive forces acting on a snake during lateral undulation is given in appendix A.5.

4.4 Dynamic analysis

4.4.1 Newton-Euler equations of motion

Using the kinematics and having models for the friction forces the Newton-Euler equations of motion can be derived. To derive the dynamic model, the forces acting on link i are shown in figure 4.3.

A snake robot with N links has N + 2 degrees of freedom. These degrees of freedom are N-1 internal 'shape' degrees of freedom plus 2 translational and 1 rotational degree of freedom of the center of mass of the snake. The resulting equation of motion of the snake robot is given by 4.6.

$$\boldsymbol{M}_\theta \ddot{\boldsymbol{\theta}} + \boldsymbol{W}\dot{\boldsymbol{\theta}}^2 - l\boldsymbol{S}_\theta \boldsymbol{K}\boldsymbol{f}_{R,x} + l\boldsymbol{C}_\theta K \boldsymbol{f}_{R,y} = \boldsymbol{D}^T \boldsymbol{u} \tag{4.6a}$$

$$Nm \begin{bmatrix} \ddot{p}_x \\ \ddot{p}_y \end{bmatrix} = \boldsymbol{E}^T \boldsymbol{f}_R \tag{4.6b}$$

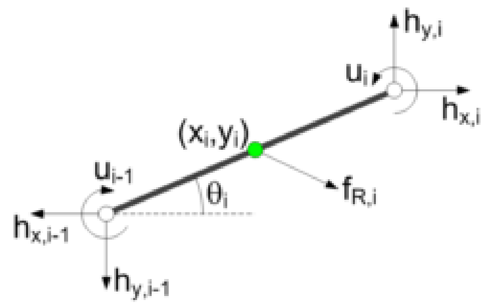


Figure 4.3: Forces acting on link i [11]

$\boldsymbol{f}_R$ is the vector of friction forces. $\boldsymbol{u}$ is the vector with torque inputs from the actuators at the hinges. This form of the equation of motion is obtained via various intermediate steps. For a detailed analysis see [11]. The equations to obtain the symbols $\boldsymbol{M_\theta}$, $\boldsymbol{W}$ and $\boldsymbol{K}$ are in appendix A.6.

From the equations of motion the internal constraint forces between the links ($\boldsymbol{h}_x$ and $\boldsymbol{h}_y$) are already removed since they internally cancel each other at the joints. From 4.6b it becomes apparent that the position of the center of mass of the snake is solely determine by the summation of all the friction forces. This is as expected. From equation 4.6a it becomes apparent that the absolute angles of the snake's links are partly influenced by the input torques ($\boldsymbol{u}$) and partly by the friction forces ($\boldsymbol{f}_R$).

A snake robot has N+2 degrees of freedom. However, it only has N-1 actuators. One actuator at every connection between two links. This results in an underactuated system. There are less actuators than degrees of freedom. This also becomes apparent from the equations of motion (equation 4.6) since the position of the robot's center of mass is influenced by the friction forces and not directly by the input torques.

4.4.2 Lagrange equations of motion

The snake robots motion can also be expressed in the Lagrange equations of motion. The kinematic analysis does not change for these equations of motion. The Lagrange equations of motion are given by 4.7.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right) - \frac{\partial L}{\partial \boldsymbol{q}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \mathbf{A}(\mathbf{q})\lambda + \mathbf{B}(\mathbf{q})\mathbf{u} \tag{4.7a}$$

$$\mathbf{A}^T(\mathbf{q})\dot{\mathbf{q}} = 0 \tag{4.7b}$$

$\boldsymbol{q}$ are the generalized coordinates of the snake robot, $\boldsymbol{A}(\boldsymbol{q})\boldsymbol{\lambda}$ are the snake robot constraint forces, $\boldsymbol{B}(\boldsymbol{q})\mathbf{u}$ are the externally applied forces and L is the Lagrangian of the system. The constraint forces are defined such that they should satisfied 4.7b.

The Lagrangian is defined as:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) - E_{pot}(\boldsymbol{q}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{4.8}$$

Where E_{kin} is the kinetic energy and E_{pot} the potential energy of the snake. In this research the snake describes planar motion so the Lagrangian is equal to the kinetic energy of the snake ($L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = E_{kin}(\boldsymbol{q}, \dot{\boldsymbol{q}})$).

The Lagrangian for a 4 link snake robot is:

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \sum_{i=1}^{4} \frac{1}{2}(m(\dot{x}_i^2(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \dot{y}_i^2(\boldsymbol{q}, \dot{\boldsymbol{q}})) + J\dot{\theta}_i^2) \tag{4.9}$$

Where $x_i(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and $y_i(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are given in appendix A.7.

4.4.3 Port-Hamiltonian equations of motion

The port-Hamiltonian approach to modelling and control of complex physical systems is a bit different from the Newton-Euler and Euler-Lagrange modelling explained so far. In port-Hamiltonian modelling a system is viewed by the energy and power flow between its components. The energy of the system is not described in term of the generalized coordinates and its derivative (positions and velocities) but in the generalized positions ($\boldsymbol{q}$) and the generalized momenta ($\boldsymbol{p}$) of the system. Port-Hamiltonian modelling can easily be extended to multiple physical domains although for the snake robot we stay in the mechanical domain only. Port-Hamiltonian modelling is useful to get a better understanding of what is physically happening in a system because of its energy and flow description. Because of this properties it is a very suitable technique to really model for control [2],[23]. The equations of motion are defined in equation 4.10.

$$\dot{\boldsymbol{q}} = \frac{\partial H}{\partial \boldsymbol{p}}(\boldsymbol{q}, \boldsymbol{p}) \tag{4.10a}$$

$$\dot{\boldsymbol{p}} = -\frac{\partial H}{\partial \boldsymbol{q}}(\boldsymbol{q}, \boldsymbol{p}) + \boldsymbol{A}(\boldsymbol{q})\lambda + \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} \tag{4.10b}$$

$$\boldsymbol{y} = \boldsymbol{B}^T(\boldsymbol{q})\frac{\partial H}{\partial \boldsymbol{p}}(\boldsymbol{q}, \boldsymbol{p}) \tag{4.10c}$$

$$\boldsymbol{0} = \boldsymbol{A}^T(\boldsymbol{q})\frac{\partial H}{\partial \boldsymbol{p}}(\boldsymbol{q}, \boldsymbol{p}) \tag{4.10d}$$

Where the generalized coordinates are $\boldsymbol{q}$ and the generalized momenta are $\boldsymbol{p}$. $\boldsymbol{A}(\boldsymbol{q})\boldsymbol{\lambda}$ are as in the Lagrange equations of motion the constraint forces and $\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}$ are the externally applied forces.

The Hamiltonian $H(\boldsymbol{q}$,$\boldsymbol{p})$ is defined as the total energy of the system (equation 4.11):

$$H(\boldsymbol{q}, \boldsymbol{p}) = E_{kin}(\boldsymbol{q}, \boldsymbol{p}) + E_{pot}(\boldsymbol{q}) \tag{4.11}$$

The Lagrangian and Hamiltonian are very alike and in case of the snake robot moving over a flat surface they are equal. Because in general the Lagrangian and Hamiltonian are very alike the Lagrange equations of motion can be transformed to the Hamiltonian equations of motion fairly easy with the use of equations 4.10 and 4.12.

$$\boldsymbol{p} = \frac{\partial L}{\partial \dot{\boldsymbol{q}}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{q}} \tag{4.12}$$

$\boldsymbol{M}(\boldsymbol{q})$ is the mass matrix of the system. The port-Hamiltonian equations of motion from equation 4.10 can be rewritten in the general input-state-output form from equation 4.13.

$$\dot{\boldsymbol{x}} = \big(\boldsymbol{J}(\boldsymbol{x}) - \boldsymbol{R}(\boldsymbol{x})\big)\frac{\partial H}{\partial \boldsymbol{x}}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u} \tag{4.13a}$$

$$\boldsymbol{y} = \boldsymbol{g}^T(\boldsymbol{x})\frac{\partial H}{\partial \boldsymbol{x}}(\boldsymbol{x}) \tag{4.13b}$$

$\boldsymbol{x} = \big[\boldsymbol{q}, \boldsymbol{p}\big]^T$. $\boldsymbol{J}(\boldsymbol{x})$ is the power-conserving internal interconnection structure of the system which connects all energy storing and energy dissipating elements. $\boldsymbol{R}(\boldsymbol{x})$ is the resistive (energy dissipating) structure. More details about the derivation of $\boldsymbol{J}(\boldsymbol{x})$ and $\boldsymbol{R}(\boldsymbol{x})$ related to the 4 link snake robot case are given in appendix A.8.

4.5 Control

For the control of the snake robot there are a few different steps and control goals that have to be distinguished from each other.

There can be many control goals depending on the implementation of the specific snake robot. Some of the control goals are to move the snake's center of mass from place A to place B and/or control the snake's shape and/or control the snake's head (which may be equipped with sensors). Depending on the application there may be different sensors installed on the snake robot. The head can be equipped with a infrared sensor and/or a CO_2 sensor. Furthermore there may be a sensor (camera or someone observing the robot) to "measure" the snake's center of mass position. However, whatever the application, the snake robot will always need sensors to measure the relative angle between its links to know the snake's shape because the snake's shape through time will determine its locomotion. The focus of this research is on designing a controller for controlling the snake's actual shape based on a desired reference shape.

4.5.1 Reference trajectory

As already mentioned in 4.1 a snake propels it's self forward via lateral undulation by sending a serpenoid curve down its body. The serpenoid curve moving down a snake's body can be discretised in N links and the resulting relative link angles required to form this serpenoid curve are given by equation 4.14 along with first and second time derivatives (useful for control purposes later on). These time derivatives (4.14b and 4.14c) are valid under the assumption that γ is a constant not depending on time.

$$\phi_{i,ref} = \alpha sin(\omega t + (i-1)\delta) + \gamma \tag{4.14a}$$

$$\dot{\phi}_{i,ref} = \alpha \omega cos(\omega t + (i-1)\delta) \tag{4.14b}$$

$$\ddot{\phi}_{i,ref} = -\alpha \omega^2 sin(\omega t + (i-1)\delta) \tag{4.14c}$$

In these equations α is the amplitude of the link angle sinusoidal motion during the locomotion. ω is the angular frequency of the periodic motion. t denotes time. δ is the phase shift between the different joint angles and γ is the bias. α,δ and γ determine the shape of the serpenoid curve and ω determines its speed. If the parameter γ is zero the snake moves in a straight line. Otherwise the snake will follow a more curved path (see appendix A.2 for the effect of changing serpenoid curve parameters on the resulting serpenoid curve).

4.5.2 Newton-Euler shape control

The control target for planar snake robot locomotion in general has two stages. First of all, to move the snake from place A to place B the snake has to send a desired serpenoid curve through its body. The center of mass will move and based on this movement the desired serpenoid curve will be changed by a controller to let the snake move from point A to B correctly. Lets call this path control. Secondly the snake robot link angles have to correspond to the desired link angles from the desired reference serpenoid curve. So the controller also has to take care of that. Lets call that shape control. This research focuses on designing a controller for the shape control in the port-Hamiltonian framework.

Pål Liljebäck already proposed a simple joint controller and exponentially stable joint controller in combination with partial feedback linearization for shape control (control of the link angles). This controller is designed in the Newton-Euler framework. For this controller it is desired to separate the equations of motion in actuated and non-actuated dynamics. Hence a new set of generalized coordinates is introduced. The new set of generalized coordinates consists of N-1 relative angles ϕ_i between two links, 1 absolute angle (of the head link of the robot) θ_N and the two (x and y) positions of the center of mass of the robot. The transformation from the absolute angels $\boldsymbol{\theta}$ to the relative angels $\boldsymbol{\phi}$ is given by $\boldsymbol{\theta} = \mathbf{H}\bar{\boldsymbol{\phi}}$. $\bar{\boldsymbol{\phi}} = \left[\phi_1, ..., \phi_{N-1}, \theta_N\right]^T \in \mathbb{R}^N$. For the definition of $\mathbf{H}$ see appendix A.1.

The new set of generalized coordinates becomes:

$$\boldsymbol{q}_\phi = \begin{bmatrix} \bar{\phi} \\ \boldsymbol{p} \end{bmatrix} \tag{4.15}$$

Applying a coordinate transformation from $\boldsymbol{q}$ to $\boldsymbol{q}_\phi$ and pre-multiplying equation 4.6a with $\boldsymbol{H}^T$ gives the decoupled equations of motion. The decoupled equations of motion are in equation 4.16.

$$\bar{\boldsymbol{M}}(\bar{\boldsymbol{\phi}})\ddot{\boldsymbol{q}}_{\boldsymbol{\phi}} + \bar{\boldsymbol{W}}(\bar{\boldsymbol{\phi}}, \dot{\bar{\boldsymbol{\phi}}}) + \bar{\boldsymbol{G}}(\bar{\boldsymbol{\phi}})\boldsymbol{f}_R(\bar{\boldsymbol{\phi}}, \dot{\bar{\boldsymbol{\phi}}}, \dot{\mathbf{p}}) = \bar{\boldsymbol{B}}\boldsymbol{u} \tag{4.16}$$

The equations for $\bar{\boldsymbol{M}}$,$\bar{\boldsymbol{W}}$,$\bar{\boldsymbol{G}}$ and $\bar{\boldsymbol{B}}$ are in appendix A.9.

These equations can be rewritten to the form of equation 4.17.

$$\bar{\boldsymbol{M}}_{\boldsymbol{11}}\ddot{\boldsymbol{q}}_{\boldsymbol{a}} + \bar{\boldsymbol{M}}_{\boldsymbol{12}}\ddot{\boldsymbol{q}}_{\boldsymbol{u}} + \bar{\boldsymbol{W}}_{\boldsymbol{1}} + \bar{\boldsymbol{G}}_{\boldsymbol{1}}\boldsymbol{f}_R = \boldsymbol{u} \tag{4.17a}$$

$$\bar{\boldsymbol{M}}_{\boldsymbol{21}}\ddot{\boldsymbol{q}}_{\boldsymbol{a}} + \bar{\boldsymbol{M}}_{\boldsymbol{22}}\ddot{\boldsymbol{q}}_{\boldsymbol{u}} + \bar{\boldsymbol{W}}_{\boldsymbol{2}} + \bar{\boldsymbol{G}}_{\boldsymbol{2}}\boldsymbol{f}_R = \mathbf{0}_{3x1} \tag{4.17b}$$

where $\boldsymbol{q}_{\boldsymbol{a}} = \left[\phi_1, ..., \phi_{N-1}\right]^T \in \mathbb{R}^{N-1}$, $\boldsymbol{q}_{\boldsymbol{u}} = \left[\theta_N, p_x, p_y\right]^T \in \mathbb{R}^3$, $\bar{\boldsymbol{M}}_{\boldsymbol{11}} \in \mathbb{R}^{(N-1)\times(N-1)}$, $\bar{\boldsymbol{M}}_{\boldsymbol{12}} \in \mathbb{R}^{(N-1)\times 3)}$, $\bar{\boldsymbol{M}}_{\boldsymbol{21}} \in \mathbb{R}^{3\times(N-1)}$, $\bar{\boldsymbol{M}}_{\boldsymbol{22}} \in \mathbb{R}^{3\times 3}$, $\bar{\boldsymbol{W}}_{\boldsymbol{1}} \in \mathbb{R}^{(N-1)}$, $\bar{\boldsymbol{W}}_{\boldsymbol{2}} \in \mathbb{R}^3$, $\bar{\boldsymbol{G}}_{\boldsymbol{1}} \in \mathbb{R}^{(N-1)\times 2N}$ and $\bar{\boldsymbol{G}}_{\boldsymbol{2}} \in \mathbb{R}^{3\times 2N}$. In the uncoupled equations of motion $\boldsymbol{q}_{\boldsymbol{a}}$ represent the actuated degrees of freedom where the $\boldsymbol{q}_{\boldsymbol{u}}$ represent the unactuated degrees of freedom.

Since we are dealing with an underactuated systems, we cannot apply a full feedback linearization by adding the inverse dynamics to the actuator inputs. However, a partial feedback linearization is possible. The partial feedback linearization will cancel all the modelled (highly-nonlinear) counteracting friction forces on the hinges. This is done by solving equation 4.17b for $\ddot{\boldsymbol{q}}_{\boldsymbol{u}}$ and substituting it in 4.17a. After that, equation 4.17a can be solved for $\boldsymbol{u}$. Replacing $\ddot{\boldsymbol{q}}_{\boldsymbol{a}}$ with $\bar{\boldsymbol{u}}$ gives the linearizing controller as presented in equation 4.18.

$$\boldsymbol{u} = (\bar{\boldsymbol{M}}_{\boldsymbol{11}} - \bar{\boldsymbol{M}}_{\boldsymbol{12}}\bar{\boldsymbol{M}}_{\boldsymbol{21}}{}^{-1}\bar{\boldsymbol{M}}_{\boldsymbol{21}})\bar{\boldsymbol{u}} + \bar{\boldsymbol{W}}_{\boldsymbol{1}} + \bar{\boldsymbol{G}}_{\boldsymbol{1}}\boldsymbol{f}_R - \bar{\boldsymbol{M}}_{\boldsymbol{12}}\bar{\boldsymbol{M}}_{\boldsymbol{22}}{}^{-1}(\bar{\boldsymbol{W}}_{\boldsymbol{2}} + \bar{\boldsymbol{G}}_{\boldsymbol{2}}\boldsymbol{f}_R) \tag{4.18}$$

This control input gives us the mapping $\ddot{\boldsymbol{q}}_{\boldsymbol{a}} = \bar{\boldsymbol{u}}$.

The desired (reference) angles of the snake's links are know (equation 4.14a) and applying a simple joint controller or exponentially stable joint controller (for the controller equation see appendix A.10) gives the required $\bar{\boldsymbol{u}}$. Applying equation 4.18 than determines the desired torque to get the required joint accelerations $\ddot{\boldsymbol{q}}_{\boldsymbol{a}}$.

4.5.3 port-Hamiltonian shape control

Besides a shape controller in the Newton-Euler framework a controller can also be designed in the port-Hamiltonian framework. This framework should make controller design more convenient. The passivity based Energy-Casimir method is applied for shape control of the snake robot. When the resistive structure property $\boldsymbol{R}(\boldsymbol{x}) \geq 0$ and energy conserving property of $\boldsymbol{J}(\boldsymbol{x}) = -\boldsymbol{J}^T(\boldsymbol{x})$ are satisfied the Hamiltonian of the port-Hamiltonian system is a stable Lyapunov function ($\frac{dH}{dt} \leq 0$). In the Energy-Casimir method the port-Hamiltonian system is expanded with a Casimir function from which the control law is derived. This functions must be a positive (semi) definite energy-like function. The modified Hamiltonian is chosen as $H_{mod} = H + C$. To guarantee stability of the modified system the Lyapunov of the modified system should also be stable which is guaranteed if $\frac{dC}{dt} = 0$. There can be proven that a suitable Casimir function for a port-Hamiltonian system with dissipation satisfies 4.19 [2], [23]. Only Casimirs that do not contain derivatives nor primitives of the states (of the system) satisfied this criterion.

$$\frac{d}{dt}\boldsymbol{C}(\boldsymbol{x}(t)) = \frac{\partial^T \boldsymbol{C}}{\partial \boldsymbol{x}}(\boldsymbol{x}(t))\dot{\boldsymbol{x}}(t) = 0 \tag{4.19}$$

Finding $\frac{dC}{dt} = 0$ is finding the equilibrium point of the Casimir. The shape control problem in this research is a tracking problem between the actual and desired link angle. The control problem comes down to bringing the tracking error to (equilibrium) zero. Therefore the Casimir states are expressed in the tracking error $\tilde{\boldsymbol{x}}$. Again a coordinate transformation to the actuated and unactuated generalized coordinates ($\boldsymbol{q_a}$ and $\boldsymbol{q_u}$ see section 4.5.2) is needed. The control laws are only obtained for the (N-1) actuated degrees of freedom. This results in 3 Casimirs, one for every actuator. In appendix A.11 the coordinate transformation of the port-Hamiltonian input-state-output representation to the actuated and unactuated coordinates is given. Incorporating the coordinate transformation in the (tracking error) states of the Casimirs results in the applied Casimirs being expressed in $\tilde{x}_i = \phi_{i,ref} - \phi_i$.

The resulting control laws are obtained by solving 4.19 for u_1 till u_3.

In this research multiple Casimirs are tested to find a suitable control law. The Casimirs are expressed in the error states $\tilde{x}_{q,i} = \phi_{q,i,ref} - \phi_{q,i}$, $\tilde{x}_{p,i} = \phi_{p,i,ref} - \phi_{p,i}$, $\dot{\tilde{x}}_{q,i} = \dot{\phi}_{q,i,ref} - \dot{\phi}_{q,i}$ and $\dot{\tilde{x}}_{p,i} = \dot{\phi}_{p,i,ref} - \dot{\phi}_{p,i}$. Among others the following three Casimirs are used:

$$C_i = \frac{1}{2}a_i(\tilde{x}_{q,i})^2 + \frac{1}{2}b_i(\tilde{x}_{p,i})^2 \tag{4.20a}$$

$$C_i = \frac{1}{2}(a_i\tilde{x}_{q,i} + b_i\tilde{x}_{p,i})^2 \tag{4.20b}$$

$$C_i = a_i\tilde{x}_{q,i}\tilde{x}_{p,i} \tag{4.20c}$$

Since $\boldsymbol{q_a} \in \mathbb{R}^{(N-1)}$ $i = 1, 2, 3$.

Equation 4.19 combined with the first Casimir (equation 4.20a) gives:

$$\frac{dC_i}{dt}\dot{x} = \begin{bmatrix} a_1(\phi_{q,1,ref} - \phi_{q,1})(\dot{\phi}_{q,1,ref} - \dot{\phi}_{q,1}) + b_1(\phi_{p,1,ref} - \phi_{p,1})(\dot{\phi}_{p,1,ref} - \dot{\phi}_{p,1} \\ a_2(\phi_{q,2,ref} - \phi_{q,2})(\dot{\phi}_{q,2,ref} - \dot{\phi}_{q,2}) + b_2(\phi_{p,2,ref} - \phi_{p,2})(\dot{\phi}_{p,2,ref} - \dot{\phi}_{p,2} \\ a_3(\phi_{q,3,ref} - \phi_{q,3})(\dot{\phi}_{q,3,ref} - \dot{\phi}_{q,3}) + b_3(\phi_{p,3,ref} - \phi_{p,3})(\dot{\phi}_{p,3,ref} - \dot{\phi}_{p,3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.21}$$

Where (for mechanical systems) the unknown (to solve for) torques and friction forces on the hinges enter the equation via the $\dot{\phi}_{p,i}$ term (see appendix A.11). Therefore only Casimirs which have $\dot{\phi}_{p,i}$ in $\frac{dC_i}{dt} = 0$ are useful in the shape controller design. Ideally the obtained controller should not cancel but use the friction forces and non-linearity of the system for controlling the snake robot. Port-Hamiltonian control exactly tries to do that by redirecting energy flows in the system instead of cancelling them with input forces.

The system of equations like equation 4.21 for the second and third Casimir are given in appendix A.12. Solving the systems of equations for the different Casimirs gives the different control laws for the hinge actuators. The resulting control laws for the different Casimirs are given in appendix A.13.

The control law for u_1 obtained from the first Casimir (equation 4.20a) is a fraction with in the numerator a multiplication of the terms $(\phi_{p,1} - \phi_{p,1,ref}), (\phi_{p,2} - \phi_{p,2,ref})$ and $(\phi_{p,3} - \phi_{p,3,ref})$. These terms can become zero (which they should in proper tracking). Because of these terms the control law can (and will) become singular so it is not suitable. The numerator for u_2 and u_3 also contain the same multiplication terms as u_1.

The system of equations resulting from $\frac{dC_i}{dt}$ for the first and third Casimir (equation 4.21 and A.19) are very comparable. This results in their control laws also being very comparable (see appendix A.9). The control law for u_1 obtained form the third Casimir (equation 4.20c) contains a multiplication of the terms $(\phi_{q,1} - \phi_{q,1,ref}), (\phi_{q,2} - \phi_{q,2,ref})$ and $(\phi_{q,3} - \phi_{q,3,ref})$ in its numerator as well. Also these terms can and will become zero resulting in a singularity in the control law and therefore a non suitable control law. Again u_2 and u_3 also contain the same multiplication terms as u_1.

The control laws from the second Casimir are worth looking into.

5 Method

From the analysis part it becomes clear that there are multiple modelling techniques and frameworks in which equations of motions and controllers can be derived. The solution path for this research consists of several steps. First of all the dynamics of and controller for the snake robot's motion are derived via the Newton-Euler established method to understand the robots dynamics better. The result is implemented and simulated in both MATLAB and Simulink. The results of the simulations are compared to what is expected from literature. Further more the results of both simulation in MATLAB and Simulink should be (approximately) the same. If both these checks give a positive result than the first step in the modelling and simulation is correctly implemented and the first research goal is met.

For the second step the Euler-Lagrange equations of motion are derived and than implemented and simulated in MATLAB. The resulting motion of the snake should (with the same input) be corresponding to the motion of the snake obtained from the Newton-Euler equations of motion simulation in MATLAB. If this is the case than the second research goal is also met.

From the Euler-Lagrange equations of motion the step to the port-Hamiltonian equations of motions will be made and the resulting equations are implemented and simulated in MATLAB. The resulting motion should (with the same input) be corresponding to the results obtained from the Newton-Euler and Lagrange equations of motion. If the results are corresponding the third research goal is met as well.

Finally a port-Hamiltonian controller will be designed to exploit the benefits of port-Hamiltonian modelling. The controller will first be designed theoretically and after that it will be implemented and simulated in MATLAB. From comparing the resulting motion to the desired (reference) motion the performance of the controller will be studied and compared to the controller derived in the Newton-Euler framework.

This specific solution route is chosen because it is convenient and because the port-Hamiltonian equations of motion and port-Hamiltonian controller are new for a snake robot.

To summarize the method of this research:

1. Derive the equations of motion and controller for a snake robot in the Newton-Euler framework. Implement and simulate the results in MATLAB and Simulink
   (a) Compare the results from MATLAB with expected from literature
   (b) Compare the results from MATLAB and Simulink to each other
2. Derive, implement and simulate (in MATLAB) the Euler-Lagrange equations of motion for a snake robot
   (a) Compare the Lagrange equation of motion results to the Newton-Euler results
3. Derive, implement and simulate (in MATLAB) the port-Hamiltonian equations of motion for a snake robot
   (a) Compare the port-Hamiltonian equation of motion results to the Lagrange results
4. Design and implement a port-Hamiltonian controller
   (a) Compare the resulting motion of the snake's shape to the reference shape and determine the performance of the controller

The numbers of the method steps (1 till 4) correspond to research goals 1 till 4.

5.1 Method step 1

5.1.1 MATLAB implementation and simulation

In this section the MATLAB code applied to derive and simulate the equations of motion and controller in the Newton-Euler framework is enumerated in pseudocode.

1. Initialize parameters snake robot (m,l,N), simulation parameters (simulation steps, simulation time), reference input parameters (α,ω,δ,γ) and controller parameters (k_p,k_d)
2. Initialize calculation matrices $\mathbf{A}$,$\mathbf{D}$,$\mathbf{K}$,$\mathbf{V}$,$\mathbf{H}$,$\mathbf{e}$,$\mathbf{E}$ (appendix A.1) and initial configuration snake robot $\boldsymbol{q(0)}$
3. Determine equation of motion matrices $\bar{\boldsymbol{M}}$,$\bar{\boldsymbol{W}}$,$\bar{\boldsymbol{G}}$ and $\bar{\boldsymbol{B}}$ (equation 4.16) symbolically
4. Determine the friction force equations (equation 4.4) symbolically
5. Starting simulation loop:
   (a) Determine required input torques ($\bar{\boldsymbol{u}}$) with simple joint controller or exponentially stable controller (appendix A.10)
   (b) Determine required $\mathbf{u}$ (equation 4.18).
   (c) Determine resulting $\ddot{\boldsymbol{q}}_{\boldsymbol{\phi}}$ (equation 4.16).
   (d) Obtain $\dot{\boldsymbol{q}}_{\boldsymbol{\phi}}$ and $\boldsymbol{q}_{\boldsymbol{\phi}}$ via forward Euler time integration
   (e) Obtain new states of the snake and increment time till the simulation time is over

Via those steps the motion of the snake robot is simulated in MATLAB and the results for different inputs and system parameters are saved.

5.1.2 Simulink implementation and simulation

The same Newton-Euler equations of motion and controller are implemented in Simulink to simulate the snake robot motion. The resulting block scheme is shown in figure 5.1. From this block diagram the complex feedback between the snake's movement of the body and the (propelling) friction forces becomes clear.

5.2 Method step 2

For the derivation and simulation of the Lagrange equation of motion the general formula from equation 4.7 is used. In the specific application of this research the external forces (input torques and friction forces) are transformed to generalized input forces. Hence the term $\boldsymbol{B}(\boldsymbol{q})\boldsymbol{u}$ becomes a term containing only generalized input forces ($\boldsymbol{U}_{ext}$). This means that there are no constraint forces ($\boldsymbol{A}(\boldsymbol{q})\boldsymbol{\lambda} = 0$) because of the generalized inputs. $\boldsymbol{U}_{ext}$ is calculated with equation 5.1.

$$\boldsymbol{U_{ext}} = \begin{bmatrix} l\boldsymbol{S}_\theta \mathbf{K}\boldsymbol{f_{R,x}} - l\boldsymbol{C}_\theta \mathbf{K}\boldsymbol{f_{R,y}} + \mathbf{D}^T\mathbf{u} \\ \mathbf{E}^T\boldsymbol{f_R} \end{bmatrix} \tag{5.1}$$

The MATLAB code to derive and simulate the equations of motion of the snake robot with Lagrange is very comparable to the code used for Newton-Euler. The pseudocode for the Lagrange equations of motion is obtained by replacing the following lines of the pseudocode of the Newton-Euler simulation:

3. Determine equation of motion vector terms $\frac{d}{dt}(\frac{\partial L}{\partial \dot{q}})$, $\frac{\partial L}{\partial q}$ and U_{ext} from equation 4.7 combined with 4.8 and from 5.1.
5. (b) Load $\boldsymbol{u}$ from the Newton-Euler MATLAB simulation.

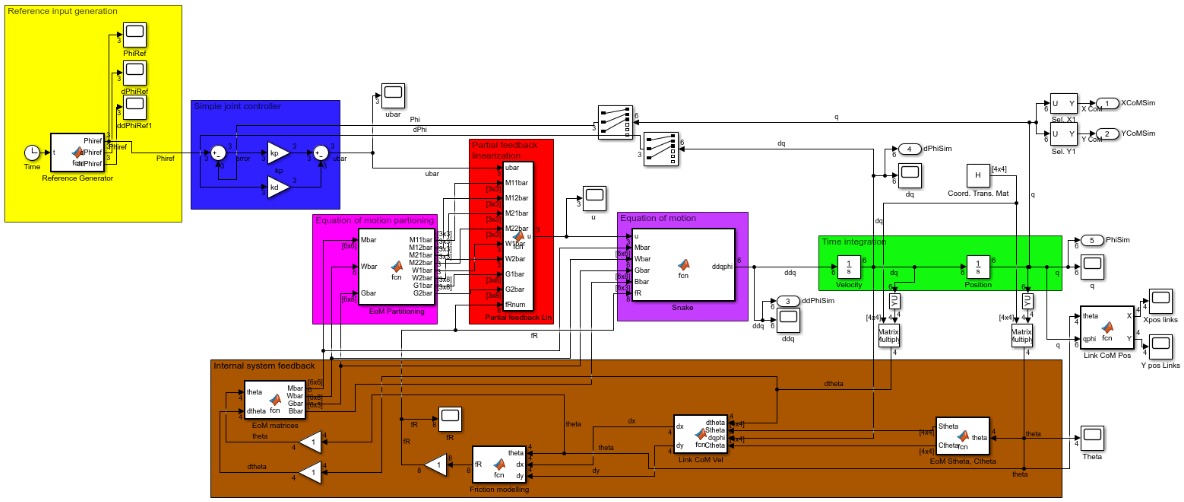


Figure 5.1: Simulink block diagram of snake robot. Yellow block: Reference input generation, Blue block: (Simple joint) controller, Red block: Partial feedback linearization, Pink block: Equation of motion partitioning (see equation 4.17), Purple block: Equation of motion snake, Green block: Time integration, Brown block: Internal system feedback

(c) Determine resulting $\ddot{\boldsymbol{q}}$ with equation 4.7

(d) Obtain $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$ via forward Euler time integration

With these small changes to the simulation code the Lagrange equations of motion are derived and simulated.

5.3 Method step 3

The derivation and simulation of the snake robot locomotion in the port-Hamiltonian framework requires a bit more attention because of the high non-linearity of the system. To obtain the steps for the port-Hamiltonian modelling the following lines in the Newton-Euler pseudo code are changed:

3. Determine equation of motion vector terms $\frac{\partial H}{\partial \mathbf{p}}, \frac{\partial H}{\partial \mathbf{q}}$ and $\boldsymbol{U_{ext}}$ from equation 4.10 combined with 4.11 and equation 5.1.

5. (c) Determine $\dot{\mathbf{p}}$ and $\dot{\mathbf{q}}$ with equation 4.10

   (d) Determine $\mathbf{q}$ with forward Euler time integration. Determine $\mathbf{p}$ with equation 4.12

The term $\boldsymbol{A}(\boldsymbol{q})\boldsymbol{\lambda}$ in equation 4.10 is zero just like in the Lagrange equations of motion. Furthermore the relation between the momenta of the generalized coordinates ($\mathbf{p}$) and the generalized velocities $\dot{\boldsymbol{q}}$ (equation 4.12) is highly non-linear for $\boldsymbol{q}(1{:}4)$ (θ_1 till θ_4). The $\boldsymbol{M}(\boldsymbol{q})$ term contains a lot of *sin* and *cos* terms. It is not of the form $\mathbf{p} = C\dot{q}$. That is why not forward Euler but equation 4.12 has to be used to obtain $\mathbf{p}$ in the port-Hamiltonian simulation.

The equation for $\dot{\boldsymbol{p}}$ contains the influence of $\boldsymbol{U_{ext}}$. This equation can however not be used to determine $\mathbf{p}$ (via forward Euler time integration). Therefore to still have the influence of $\boldsymbol{U_{ext}}$ in the equation of motion $\dot{\boldsymbol{q}}$ values from the Lagrange equation of motion are used. Besides these changes the rest of the simulation code is the same.

5.4 Method step 4

To simulate the performance of the designed port-Hamiltonian controller most steps in the simulation are the same as those for the simulation of the Lagrange equation of motion. The only difference is the generation of the required $\boldsymbol{u}$. The change in the Lagrange pseudocode becomes:

5. (c) Determine required $\boldsymbol{u}$ by solving equation 4.21, A.18 or A.19 depending on the chosen Casimir.

To determine the required $\boldsymbol{u}$ $\phi_{q,i,ref}$, $\dot{\phi}_{q,i,ref}$ and $\phi_{p,i,ref}$, $\dot{\phi}_{p,i,ref}$ are determined from the reference signal. To determine $\phi_{p,i,ref}$ and $\dot{\phi}_{p,i,ref}$ the reference signal in combination with equation 4.12 is used. $\phi_{q,i}$ and $\phi_{p,i}$ follow from the measured angles of the snake. To determine $\phi_{p,i}$ the measured angels (and angular velocities) in combination with equation 4.12 are used. $\dot{\phi}_{q,i}$ and $\dot{\phi}_{p,i}$ follow from the input-state-output port-Hamiltonian system from equation 4.13 combined with the measured states $\boldsymbol{x}$.

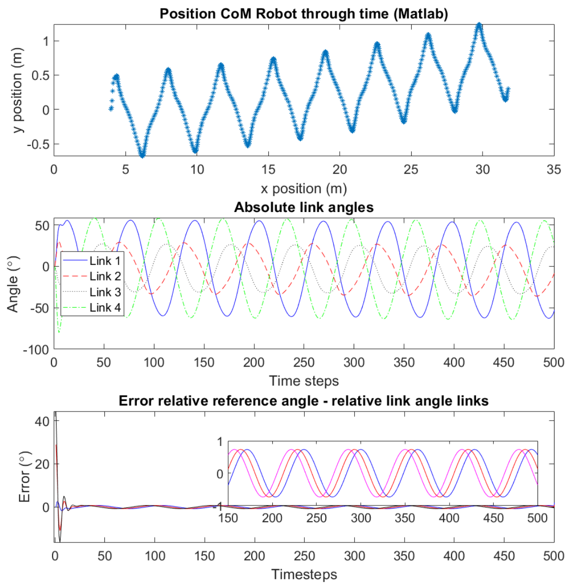


Figure 6.1: Reference simulation center of mass position, absolute link angels and joint tracking error. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple joint controller

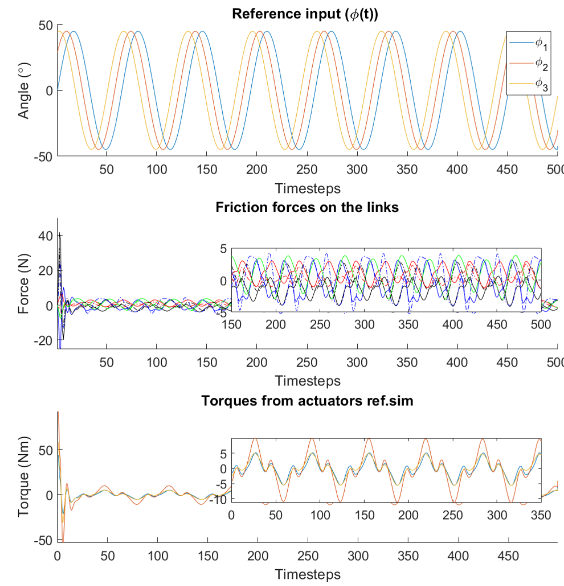


Figure 6.2: Reference simulation reference input, friction forces and actuator torques. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple joint controller

6 Results and Discussion

The results of the different method steps to reach the different research goals are listed in the following subsections. Unless otherwise indicated in the caption or legend of the figures with the simulation results the simulations are done for a $N = 4$ link snake robot with link mass $m = 1kg$ and (half) link length $l = 1m$. The tangential friction coefficient c_t is set to 1 and the normal friction coefficient c_n to 10. The simulation time is 40 seconds and the amount of simulation steps is 500. In the figures 6.1 till 6.9 the results from different simulations are plotted.

6.1 Results and Discussion step 1

6.1.1 Matlab implementation and simulation

The simulation results from the Newton-Euler equations simulated in MATLAB are compared to what can be expected (from literature).

In figure 6.1 and 6.2 the results for the reference simulation are given. All the other simulations are simulations with slightly different parameters. These are compared to the reference simulation. In this reference simulation the snakes center of mass moves in an approximate straight line from the starting point $(x, y) = (4, 0)$ to approximately $(x, y) = (31, 0.3)$. The snake gets off track slightly while it should actually not move in the y-direction. This small deviation is probably caused by the error in the numerical integration scheme. The integration from $\ddot{\boldsymbol{\phi}}$ to $\boldsymbol{\phi}$ is done with a forward Euler scheme where the curved line of the input signal (equation 4.14a and figure 6.2) is approximated by a straight line. This causes the error. Since the center of mass of the snake is not measured (nor controlled) this deviation is not compensated for. Also the initial transient error between the reference angles ϕ_{ref} and the relative angles of the snake ϕ_i (bottom plot in figure 6.1) is big. This error is big because the initial (configuration) states of the snake are put as $\boldsymbol{q} = \left[0, 0, 0, 0, 4, 0\right]^T$ and $\dot{\boldsymbol{q}} = \left[0, 0, 0, 0, 0, 0\right]^T$ which does not correspond with the first values of the

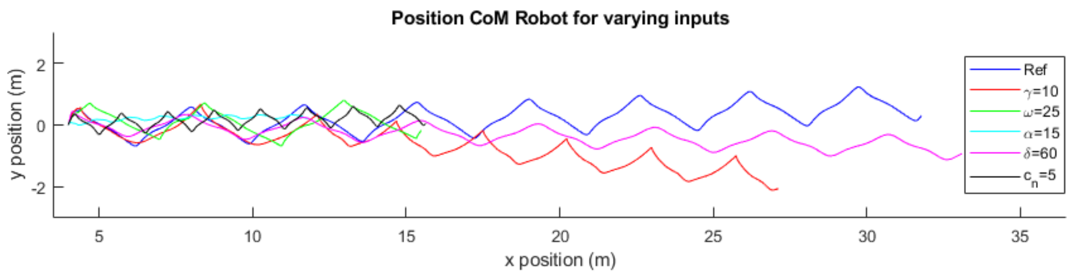


Figure 6.3: Described path by the snake robot's center of mass for different reference inputs. Simulation settings (unless otherwise indicated in the figure legend): $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple joint controller.

reference input at $t = 0$ (see equations 4.14a, 4.14b and figure 6.2). These hypothesis are confirmed from the simulations done with an exponentially stable joint controller instead of a simple joint controller. In figure 6.4 the motion of the snake robot is plotted for control with the exponentially stable joint controller instead of the simple joint controller. Further more the initial states of the snake robot are changed to $\boldsymbol{q} = \left[\theta_{1,ref}, \theta_{2,ref}, \theta_{3,ref}, 0, 4, 0\right]^T$ and $\dot{\boldsymbol{q}} = \left[\dot{\theta}_{1,ref}, \dot{\theta}_{2,ref}, \dot{\theta}_{3,ref}, 0, 0, 0\right]^T$. Changing the initial state of the snake robot results in an initial transient error of zero (bottom plot in figure 6.4).

Combining the result from partial feedback linearization $\ddot{\boldsymbol{\phi}} = \bar{\boldsymbol{u}}$ with the control law of the exponentially stable joint controller (appendix A.10) gives the resulting error dynamics $(\ddot{\phi}_{ref} - \ddot{\phi}) + k_d(\dot{\phi}_{ref} - \dot{\phi}) + k_p((\phi_{ref} - \phi) = 0$. The steady state resulting tracking error with exponentially stable controller is the bottom plot of figure 6.4. This error is much smaller than the tracking error with the simple joint controller (compare the bottom plot from figure 6.1 with the blue line in the bottom plot of figure 6.4). The resulting tracking

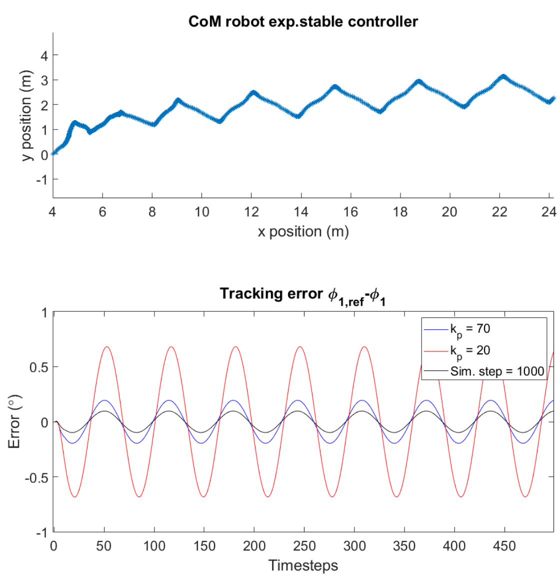


Figure 6.4: Center of mass position and joint tracking error snake with exponentially stable joint controller. Errors are only shown for ϕ_1 for clarity. Simulation settings (unless otherwise indicated in the legend): $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, exponentially stable controller

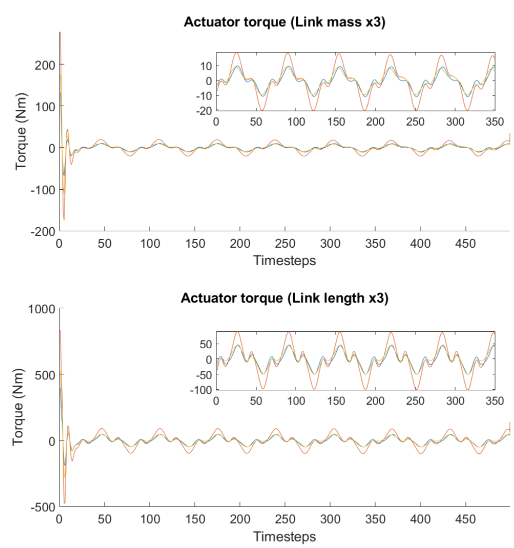


Figure 6.5: Required actuator torque for increased link masses and lengths. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple controller

errors however are not (as expected) zero. From figure 6.4 follows that decreasing the time steps within the simulation decreases the integration error and therefore also decreased the steady-state tracking error. The resulting tracking errors in figure 6.4 are therefore explained by the time integration (forward Euler) scheme. Further more lowering the controller gain k_p to 20 should increase the tracking error which it does. This supports the claim that the resulting tracking error is caused by numerical integration errors. Implementing the exponentially stable joint controller supports the hypothesis that initial configuration differences and time integration errors cause tracking errors and slight path deviations of the snake. The implemented model therefore can be assumed correct. The fact that the center of mass of the snake travels further in the reference simulation compared to the simulation with exponentially stable controller does not means that the simple controller is better. The tracking error is more important because if you want to change the distance the snake travels you change the reference input and you don't rely on the uncertain tracking errors (bigger for the simple joint controller) to make the snake travel more distance.

Comparing figure 6.3 to the reference simulation in figure 6.1 it becomes clear that changing the parameter γ (as expected from literature) causes the snake to deviate from a straight path and describe a curved trajectory instead of straight path. From figure 6.3 it becomes clear that decreasing the parameter ω indeed slows down the movement of the snake as expected from literature. Also as expected follows from figure 6.3 that decreasing the parameter α does decrease the speed of the snake robot.

Pål Liljebäck claims that a δ value of around 60° is optimal for a 4 links snake robot. Comparing figure 6.3 to 6.1 indeed shows that an angle of 60° for δ results in more forward velocity than an angle of 40°. From literature it is expected that a lower ratio $\frac{c_n}{c_t}$ will decrease the propulsive force. Figure 6.3 supports this conclusion since the snake travels less far compared the reference simulation.

Increasing the masses and length's of the links of the snake robot results (as expected) in an increase of the required torque to propel the snake robot. The conclusion is drawn from comparing figure 6.5 to the torque required in the reference simulation (figure 6.2) . Furthermore increasing the length of the links has more influence on the required torque than increasing the mass. This is as expected since the inertial for the links is calculated via $J = \frac{1}{3}ml^2$.

6.1.2 Simulink implementation and simulation

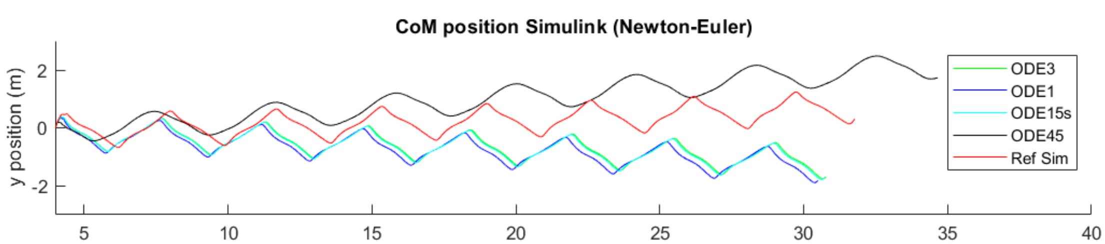


Figure 6.6: Described path by the snake robot's center of mass resulting from Simulink simulations with different solvers. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple joint controller.

Comparing the results from 6.6 with the reference simulation from figure 6.1 it becomes clear that the results are comparable but not the same. The different results are probably caused by the difference in time integration schemes used in MATLAB (a simple fixed step size forward Euler) and those used in Simulink. In Simulink the variable step size ODE45 (Dormand-Prince), variable step size ODE15s (stiff/NDF), fixed step size ODE3 (Bogacki-Shampine) and fixed step size ODE1 (Euler) are used. The difference in direction between the snake center of mass paths can be explained (as also mentioned before) because there is no feedback control for the snake's center of mass when it diverges from the intended horizontal straight line. (Initial) numerical direction errors (differing between the integration schemes) are thus not compensated for. ODE1 (Euler) is the integration scheme closest related to the forward Euler scheme used in the MATLAB simulations. However, this integration scheme also diverges significantly from the references simulation.

While conducting different simulations in Simulink it turned out that for a reference input with a too high ω the Simulink results diverged unacceptably compared to the MATLAB simulations. This phenomenon could not be explained exactly although it maybe have to do with the algebraic loops and inverse calculations in the Simulink model.

6.2 Results and Discussion step 2

In figure 6.7 the motion of the snake robot is plotted obtained from the Lagrange equations of motion. Comparing this figure to the results in figure 6.1 it becomes clear that the Lagrange equations of motion at first sight produce the same snake motion as the Newton-Euler equations of motion (as it should). However, at the bottom of figure 6.7 the error between the Lagrange and Newton-Euler simulation is plotted. The error at timestep 2 is a numerical error. This error increases with time. For the Lagrange simulations the input torques from the Newton-Euler simulation are used to check the simulations. Therefore there is not compensated for the (initial) numerical error during the simulation since the input torques are derived from the snake states from the Newton-Euler simulations. Because the system is highly non-linear and has various feedback loops (see figure 5.1) this initial error after a while increases to non-negligible values. However from the simulation results there can still be concluded that the Lagrange equations of motion are implemented correctly. First of all, the error in the simulation and its behaviour can be explained by the initial numerical error. Secondly, from figure 6.7 and 6.8 it becomes clear that (except for a numerical error) the Lagrange and port-Hamiltonian simulation give the same results and same error behaviour with respect to the Newton-Euler simulation. This confirms that the Lagrange (and port-Hamiltonian) equations of motions are implemented correctly and that the error with the Newton-Euler simulation is not related to incorrect implementation.

6.3 Results and Discussion step 3

As indicated in section 5.3 simulating the motion of the snake in the port-Hamiltonian framework is not as straight forward as in the Lagrange framework. However the correctness of $\frac{\partial H}{\partial p}$ and $\frac{\partial H}{\partial q}$ can still be checked. In figure 6.7 $\boldsymbol{q}$ obtained via Lagrange and via (time integration of) $\dot{\boldsymbol{q}} = \frac{\partial H}{\partial p}$ are compared and they turn out to be the same (apart from some numerical errors see figure 6.8). Since for both the port-Hamiltonian and the Lagrange simulation the same input torque from the Newton-Euler simulation are used both simulations show the same deviation from the Newton-Euler simulation (figure 6.7).

To check if $\frac{\partial H}{\partial q}$ is correct is harder since there are no other simulations to check the obtained values with. However if $\frac{\partial H}{\partial p}$ is correct it is also very likely that $\frac{\partial H}{\partial q}$ is correct since the only difference is the variable where H is differentiated to. Furthermore $\dot{\boldsymbol{p}} = -\frac{\partial H}{\partial q} + \boldsymbol{U_{ext}}$ holds. The $\frac{\partial H}{\partial q}$ term contains the damping effect of the energy-storing elements in the system. The snake has only moving masses as energy storing elements. This expected damping effect is very clearly visible when comparing the plot of $\dot{\boldsymbol{p}}$ to that of $\boldsymbol{U_{ext}}$ in figure 6.8. Therefore there is a very high likelihood that $\frac{\partial H}{\partial q}$ is correct.

6.4 Results and Discussion step 4

In figure 6.9 the results of the port-Hamiltonian controller derived from the second Casimir (equation 4.20b) are plotted. To analyse its performance it is compared to the result from the reference simulation with the simple joint controller in figure 6.1. The port-Hamiltonian controller moves the snake in an approximately straight line as required. However the deviation from a horizontally straight line (as is required) is a lot bigger with the port-Hamiltonian controller than with the simple joint controller. Further more the tracking error with the port-Hamiltonian controller is a lot bigger (and increases with time) than with the simple joint controller. Finally the delivered joint torque from the port-Hamiltonian controller is a lot bigger than that from the simple joint controller. The port-Hamiltonian controller is therefore not able to redirect the energy flows in the system as it should (see section 4.5.3). Another point worth mentioning is that the tracking

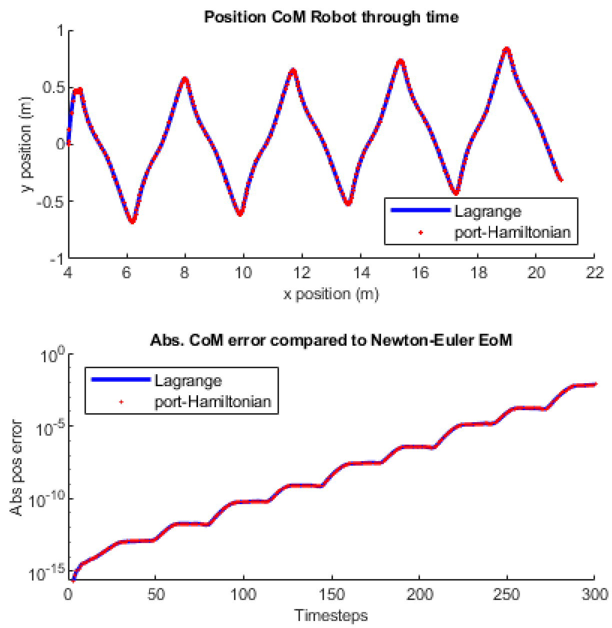


Figure 6.7: Position of the center of mass of the snake robot obtained via the port-Hamiltonian and Lagrange equations of motion in the top figure. In the bottom figure the absolute error compared to the Newton-Euler simulation is plotted. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple joint controller

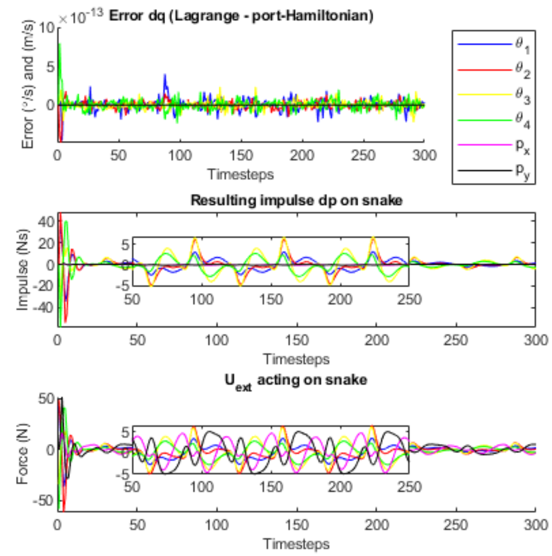


Figure 6.8: In the top figure the error is plotted for $\dot{\boldsymbol{q}}$ obtained via the port-Hamiltonian en Lagrange simulation. The middle figure contains $\dot{\boldsymbol{p}}$ obtained via the port-Hamiltonian simulation. The last figure shows $\boldsymbol{U_{ext}}$ acting on the snake during the port-Hamiltonian simulation. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, $k_p = 70$, $k_d = 5$, simple joint controller

error (and delivered torque) from the port-Hamiltonian controller are (slightly) increasing with time. The port-Hamiltonian controller probably will blow up for longer simulation times.

For the port-Hamiltonian control laws derived from the first and third Casimir it was directly clear that the control laws where not suitable for the control purpose. From figure 6.9 it becomes clear that also the second Casimir does not result in a suitable control law.

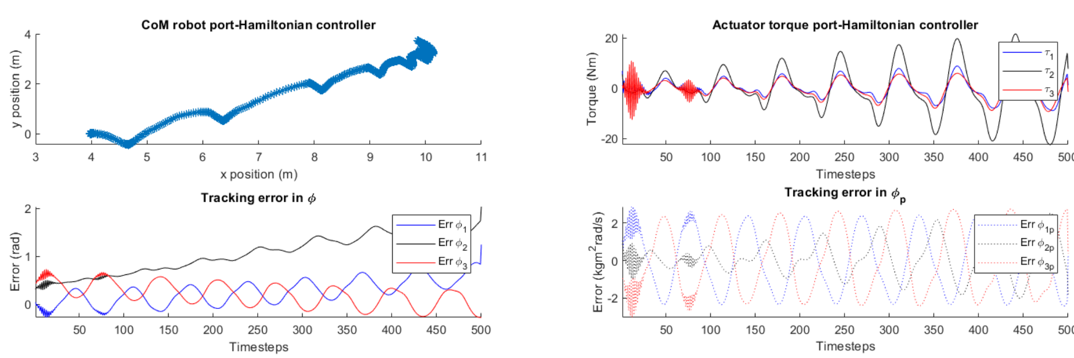


Figure 6.9: Center of mass position, actuator torques and tracking errors port-Hamiltonian controller. Simulation settings: $\alpha = 45°$,$\omega = 70°$,$\delta = 40°$,$\gamma = 0°$, Controller parameters: $a_{1,2,3} = 10$, $b_{1,2,3} = 1$, port-Hamiltonian controller.

No suitable port-Hamiltonian control law could be derived in this research. There are multiple possible explanations.

First of all, maybe the correct Casimir or the correct controller parameters ($a_{1,2,3}$ and $b_{1,2,3}$) could not be found.

Secondly the problem could also be more fundamental. As already indicated before, the snake robot is a highly non-linear underactuated system with various feedback loops. The relation between $\boldsymbol{p}$ and $\dot{\boldsymbol{q}}$ is highly non-linear and makes time simulation in the current port-Hamiltonian model difficult (explained in section 5.3). Furthermore for the tracking problem two coordinate transformations are introduced. One from the absolute to the relative degrees of freedom (section 4.5.2) and one to the error states (section 4.5.3). It could be that the combination of all these particularities make the system and control goal not suitable for the Energy-Casimir method.

7 Conclusion

From section 6.1.1 there is concluded that the equations of motion in Newton-Euler form are correctly derived and implemented. Changes in input parameters changes the output of the snake robot locomotion as expected from literature. Comparing the simulations in MATLAB with those in Simulink (section 6.1.2) there can be concluded that the time integration solver is very important in determining the motion of the snake through time. It was not possible to obtain exactly the same motion from the MATLAB and Simulink simulations. This could be due to numerical errors but the biggest differences are probably caused by the differences in the time integration (schemes).

From section 6.2 there can be concluded that the Lagrange equations of motion give the same result (except for some numerical differences) as the Newton-Euler equations of motion as long as the same time integration solver is used.

From the results from 6.3 there can be concluded that the $\frac{\partial H}{\partial p}$ is exactly correct since it gives the same results as determined in the Lagrange simulation. There is a very high likely hood that also $\frac{\partial H}{\partial q}$ is correct but this can not be concluded with absolute certainty.

From section 6.4 there can be concluded that the performance of the port-Hamiltonian controller is inferior compared to the controllers derived in the Newton-Euler framework. The Energy-Casimir method used in this research did not yield a proper control law. There are multiple possible causes mentioned in section 6.4.

Looking back at the research goals defined in section 3 there can be concluded that the first research goal is achieved although the MATLAB and Simulink models don't correspond completely. The second research goal is completely achieved. With a high certainty the third research goal is achieved as well. The last research goal could unfortunately not be achieved.

8 Recommendations

For future research it can be interesting to further study the differences between the Newton-Euler equations of motion simulated in MATLAB and those simulated in Simulink. A further understanding about how and why the different time integration solvers change the simulation outcome the way they do was not the goal of this research. However, this can be very valuable for future simulations of complex systems like the snake robots. A too high ω as reference input also causes the Simulink results to excessively diverge from the once obtained with MATLAB. Fixing the problem in the Simulink model can be beneficial for future research since the Simulink simulations are significantly faster than those conducted in MATLAB.

The most important recommendations for the future are related to the port-Hamiltonian controller design. Future research should focus on finding out why it was not possible to find a suitable port-Hamiltonian controller for this specific problem. Depending on the actual reason why the following steps could be taken:

First of all, future research could try to find a better controller parameter set. In this research various combinations are tried and the best combination resulted in the presented port-Hamiltonian controller. However not all possible controller parameter sets could be tested so this is a recommendation for the future.

Secondly it is interesting to try (more complex) Casimir functions to derive controllers from. Various Casimirs are tried in this research but again a perfect one has not yet been found.

Thirdly, if it turns out that the Energy-Casimir method is not a suitable design method for this problem the system can be altered to a dynamical control systems in port-Hamiltonian form. Instead of direct application of the Energy-Casimir method a recourse can be taken to generate Casimirs of a closed-loop system. For this the port-Hamiltonian form of the dynamical controller system is the starting point. Basically the port-Hamiltonian form of the plant and that of the controller are interconnected and from there a new controller can be derived. For more information about this method see [23].

References

[1] F. L. Chernousko. Modelling of Snake-Like Locomotions. In *Modeling, Simulation and Optimization of Complex Processes*, pages 105–114, Berlin, 2005. Springer.

[2] V. Duindam, A. Macchelli, S. Stramigioli, and H. Bruyinckx. *Modelling and control of Complex Physical Systems - The Port-Hamiltonian Approach*. Springer, Berlin, 1 edition, 2009.

[3] S. A. Fjerdingen, P. Liljebäck, and A. A. Transeth. A snake-like robot for internal inspection of complex pipe structures (PIKo). In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2009*, pages 5665–5671, St. Louis, MO, USE, 2009. IEEE.

[4] I. Grabec. Control of a creeping snake-like robot. In *7th International Workshop on Advanced Motion Control.*, pages 526–531, Maribor, Slovenia, Slovenia, 2002. IEEE.

[5] J. Gray. The mechanism of locomotion in Snakes. *Journal of Experimental Biology*, 23(2):101–120, 1946.

[6] G. P. Hicks. *Modeling and Control of a Snake-like Serial-link Structure*. Phd thesis, North Carolina State University, 2003.

[7] S. Hirose. Biologically inspired robots: snake-like locomotors and manipulators. In *2004 IEEE International Conference on Robotics and Biomimetics*, Shenyang, China, 2004. IEEE.

[8] S. Hirose and M. Mori. Biologically Inspired Snake-like Robots. In *International Conference on Robotics and Biomimetrics*, pages 1–7, 2004.

[9] D. L. Hu, J. Nirody, T. Scott, and M. J. Shelley. The mechanics of slithering locomotion. *Proceedings of the National Academy of Sciences of the United States of America*, 106(25):10081–10085, jun 2009.

[10] T. Kane and D. Lecison. Locomotion of snakes: A mechanical explanation'. *International Journal of Solids and Structures*, 37(41):5829–5837, 2000.

[11] P. Liljebäck. *Modelling, Development, and Control of Snake Robots*. PhD thesis, Norwegian University of Science and Technology, 2011.

[12] S. Ma, H. Araya, and L. Li. Development of a creeping snake-robot. In *Computational Intelligence in Robotics and Automation*, pages 77–82, Banff, Alberta, Canada, 2001. IEEE.

[13] S. Ma and N. Tadokoro. Analysis of creeping locomotion of a snake-like robot on a slope. *Autonomous Robots*, 20(1):15–23, 2006.

[14] M. H. Majid, M. R. Khan, and S. N. Sidek. Development of wheel-less snake robot with two distinct gaits and gait transition capability. *International Journal of Automation and Computing*, 10(6):534–544, 2013.

[15] H. Marvi, J. Bridges, and D. L. Hu. Snakes mimic earthworms: Propulsion using rectilinear travelling waves. *Journal of the Royal Society Interface*, 10(84):1–12, 2013.

[16] H. Marvi, G. Meyers, G. Russell, and D. L. Hu. Scalybot: A Snake-Inspired Robot With Active Control of Friction. In *Dynamic Systems and Control Conference*, pages 443–450, Miami, Florida, USA, 2012. ASME International.

[17] A. Mohammadi, E. Rezapour, M. Maggiore, and K. Y. Pettersen. Maneuvering Control of Planar Snake Robots Using Virtual Holonomic Constraints. In *IEEE Transactions on Control Systems Technology*, volume 24, pages 884–899. IEEE, may 2016.

[18] A. Morishima and S. Hirose. Design and Control of a Mobile Robot with an Articulated Body. *The International Journal of Robotics Research*, 9(2):99–114, 1990.

[19] Z. Mu, H. Wang, W. Xu, T. Liu, and H. Wang. Two types of snake-like robots for complex environment exploration: Design, development, and experiment. *Advances in Mechanical Engineering*, 9(9):1–15, 2017.

[20] M. Nilsson. Serpentine locomotion on surfaces with uniform friction. In *IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, pages 1751–1755, Sendai, Japan, 2004. IEEE.

[21] P. Prautsch and T. Mita. Control and Analysis of the gait of snake robots. In *Proc. IEEE. Int. Conf. Control Applications*, pages 502–507, Kohala Coast, HI USA, 1999.

[22] M. Saito, M. Fukaya, and T. Iwasaki. Serpentine Locomotion with Robotic Snakes. *IEEE Control Systems*, 22(1):64–81, 2002.

[23] A. Van Der Schaft. *L2-Gain and Passivity Techniques in Nonlinear Control*. Springer, Groningen, 2017.

A Appendix

A.1 Calculation matrices for modelling

The specific calculation matrices used for the modelling are defined as:

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & 1 & 1 \end{bmatrix} \qquad \mathbf{D} = \begin{bmatrix} 1 & -1 & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & 1 & -1 \end{bmatrix}$$

$$\boldsymbol{e} = \begin{bmatrix} 1, ..., 1 \end{bmatrix}^T \qquad \mathbf{E} = \begin{bmatrix} \mathbf{e} & 0_{Nx1} \\ 0_{Nx1} & \mathbf{e} \end{bmatrix}$$

$$sin\boldsymbol{\theta} = \begin{bmatrix} sin\theta_1, ..., \theta_N \end{bmatrix}^T \qquad \mathbf{S}_{\boldsymbol{\theta}} = diag(sin\boldsymbol{\theta})$$

$$cos\boldsymbol{\theta} = \begin{bmatrix} cos\theta_1, ..., \theta_N \end{bmatrix}^T \qquad \mathbf{S}_{\boldsymbol{\theta}} = diag(cos\boldsymbol{\theta})$$

$$sgn\boldsymbol{\theta} = \begin{bmatrix} sgn\ \theta_1, ..., sgn\ \theta_N \end{bmatrix}^T \qquad \dot{\boldsymbol{\theta}}^2 = \begin{bmatrix} \dot{\theta}_1^2, ..., \dot{\theta}_1^2 \end{bmatrix}^T$$

$$\boldsymbol{H} = \begin{bmatrix} 1 & 1 & 1 & ... & 1 & 1 \\ 0 & 1 & 1 & ... & 1 & 1 \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ 0 & 0 & 0 & ... & 0 & 1 \end{bmatrix} \tag{A.1}$$

Where $\mathbf{A} \in \mathbb{R}^{(N-1)\times N}$, $\mathbf{D} \in \mathbb{R}^{(N-1)\times N}$, $\mathbf{e} \in \mathbb{R}^N$, $\mathbf{E} \in \mathbb{R}^{2N\times 2}$, $sin\boldsymbol{\theta} \in \mathbb{R}^N$, $\boldsymbol{S_\theta} \in \mathbb{R}^{N\times N}$, $cos\boldsymbol{\theta} \in \mathbb{R}^N$, $\boldsymbol{C_\theta} \in \mathbb{R}^{N\times N}$,$sgn\boldsymbol{\theta} \in \mathbb{R}^N$, $\dot{\boldsymbol{\theta}}^{\mathbf{2}} \in \mathbb{R}^N$, $\boldsymbol{H} \in \mathbb{R}^{N\times N}$.

The rotation matrix from the global frame to the local frame of specific link i is defined as:

$$\boldsymbol{R}_{link,i}^{global} = \begin{bmatrix} cos\theta_i & -sin\theta_i \\ sin\theta_i & cos\theta_i \end{bmatrix} \tag{A.2}$$

σ_i is defined as:

$$\sigma_i = \begin{bmatrix} a_1, a_2, ..., a_{i-1}, \frac{a_i+b_i}{2}, b_{i+1}, b_{i+2}, ..., b_N \end{bmatrix} \tag{A.3a}$$

$$a_i = \frac{l(2i-1)}{N}, b_i = \frac{l(2i-1-2N)}{N} \tag{A.3b}$$

With $\sigma_i \in \mathbb{R}^N$.

A.2 Serpenoid curve

For lateral undulation a snake sends a serpenoid curve down its body. A serpenoid curve is a planar curve with a sinusoidally changing curvature. The x and y coordinates of a point on this curve are defined as in equation A.4.

$$x(s) = \int_0^s cos(acos(b\sigma) + c\sigma)d\sigma \tag{A.4a}$$

$$y(s) = \int_0^s sin(acos(b\sigma) + c\sigma)d\sigma \tag{A.4b}$$

The parameter s indicates the distance (of a point) on the curve from the origin. a,b and c are scalar parameters. In figure A.1 the serpenoid curve from equation A.4 for various parameter combinations is plotted.

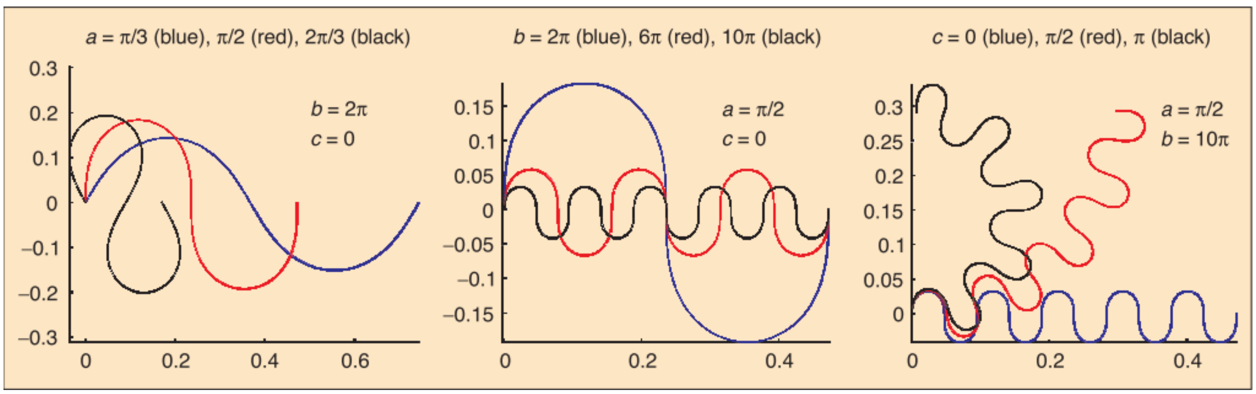


Figure A.1: Various serpenoid curves resulting from various parameter combinations for the parameters a,b and c [22]

A.3 Viscous Friction model

The viscous friction force on a specific link i in its own local coordinate frame is defined as:

$$\boldsymbol{f}_{R,i}^{link,i} = -\begin{bmatrix} c_t & 0 \\ 0 & c_n \end{bmatrix} \boldsymbol{v}_i^{link,i} \tag{A.5}$$

$\boldsymbol{f}_{R,i}^{link,i} \in \mathbb{R}^2$. $\boldsymbol{v}_i^{link,i}$ is the link velocity (in the local link coordinate frame). $link, i$ indicates the coordinate frame and $\boldsymbol{f}_{,R,i}$ indicates that it concerns the friction forces on link i. In case of anisotropic viscous friction c_t and c_n are different.

$\boldsymbol{f}_{R,i}^{link,i}$ can be transformed to the global coordinate frame with the rotation matrix (equation A.2). $\boldsymbol{f}_{R,i}^{global}$ than becomes:

$$\boldsymbol{f}_{R,i}^{global} = -\boldsymbol{R}_{link,i}^{global} \begin{bmatrix} c_t & 0 \\ 0 & c_n \end{bmatrix} (\boldsymbol{R}_{link,i}^{global})^T \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \end{bmatrix} \tag{A.6}$$

Putting the friction forces on all the links in matrix form the global frame friction forces can be written as equation 4.4 in the main report.

A.4 Coulomb Friction Model

The formula to determine the coulomb friction forces is given in equation A.7.

$$\boldsymbol{f}_R = -mg \begin{bmatrix} \mu_t \boldsymbol{C_\theta} & -\mu_n \boldsymbol{S_\theta} \\ \mu_t \boldsymbol{S_\theta} & \mu_n \boldsymbol{C_\theta} \end{bmatrix} sgn(\begin{bmatrix} \boldsymbol{C_\theta} & \boldsymbol{S_\theta} \\ -\boldsymbol{S_\theta} & \boldsymbol{C_\theta} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{X}} \\ \dot{\boldsymbol{Y}} \end{bmatrix}) \tag{A.7}$$

μ_t and μ_n are the coulomb friction coefficient in the tangential and normal direction of a link. So these are with respect to the local coordinate system of the link's. They are transformed to the global coordinate system via the rotation $\boldsymbol{R}^{global}_{link,i}$ matrix and its transpose (just as in appendix A.3). As expected from physics the friction force is opposite to the direction of motion of the links.

A.5 Visualization of propulsive forces

In figure A.2 the calculated propulsive forces acting on a snake performing lateral undulation are visualized. Snakes perform weight shifting during lateral undulation. In this research the weight shifting is neglected.

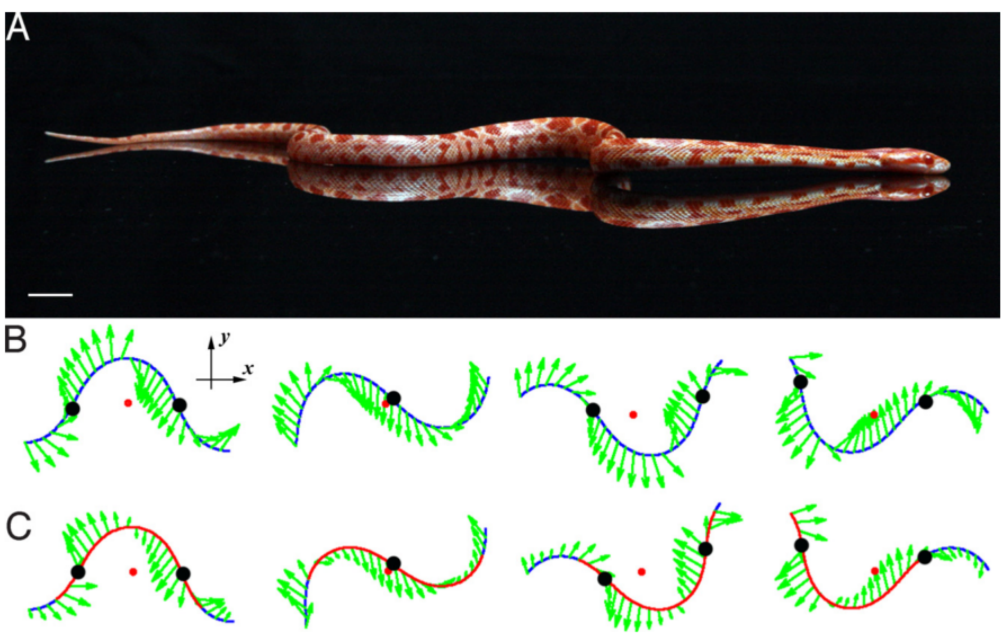


Figure A.2: A: a snake performing lateral undulation on a mirroring surface. B: Calculated propulsive forces acting on the snake (without internal weight distribution). C: Propulsive forces acting on the snake (with internal weight distribution).
The green arrows indicate the direction and size of the propulsive forces. The red lines indicate body parts with a normal force < 1. The black dots are infliction points of the body shape. The red dot is the center of mass of the snake robot. [9]

A.6 Newton-Euler modelling: symbol definitions

The symbols $\boldsymbol{M_\theta(\theta)}$, $\boldsymbol{W(\theta)}$, $\boldsymbol{K}$ and $\boldsymbol{V}$ used in the derivation of the Newton-Euler equations of motion are defined as:

$$\boldsymbol{M}_\theta(\boldsymbol{\theta}) = J\boldsymbol{I}_N + ml^2\boldsymbol{S}_\theta\boldsymbol{V}\boldsymbol{S}_\theta + ml^2\boldsymbol{C}_\theta\boldsymbol{V}\boldsymbol{C}_\theta \tag{A.8a}$$

$$\boldsymbol{W}(\boldsymbol{\theta}) = ml^2\boldsymbol{S}_\theta\boldsymbol{V}\boldsymbol{C}_\theta - ml^2\boldsymbol{C}_\theta\boldsymbol{V}\boldsymbol{S}_\theta \tag{A.8b}$$

$$\boldsymbol{K} = \boldsymbol{A}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}\boldsymbol{D} \tag{A.8c}$$

$$\boldsymbol{V} = \boldsymbol{A}^T(\boldsymbol{D}\boldsymbol{D}^T)^{-1}\boldsymbol{A} \tag{A.8d}$$

A.7 Lagrangian transformation to generalized coordinates

To express the Lagrangian in the generalized coordinates the following two equations are applied:

$$\dot{x}_i(\boldsymbol{q},\dot{\boldsymbol{q}}) = l(\sum_{j=1}^{4}(K_{i,j}^T sin(\theta_j)\dot{\theta}_j) + \dot{p}_x \tag{A.9a}$$

$$\dot{y}_i(\boldsymbol{q},\dot{\boldsymbol{q}}) = -l(\sum_{j=1}^{4}(K_{i,j}^T cos(\theta_j)\dot{\theta}_j) + \dot{p}_y \tag{A.9b}$$

Where $\boldsymbol{K}$(i,j) is the ith row and jth column of the transposed $\boldsymbol{K}$ matrix.

A.8 port-Hamiltonian Input-State-Output representation snake robot

For the vectors and matrices dimensions of the port-Hamiltonian input-state-output equations from 4.13 holds the following. Since $\boldsymbol{q} \in \mathbb{R}^{N+2}$ and $\boldsymbol{p} \in \mathbb{R}^{N+2}$ $\boldsymbol{J}(\boldsymbol{x}), \boldsymbol{R}(\boldsymbol{x})$ are $\in \mathbb{R}^{2(N+2)X2(N+2)}$. Furthermore the actuator input force vector $\boldsymbol{g}(\boldsymbol{x}) \in \mathbb{R}^{2(N+2)XN-1}$ since $\boldsymbol{u} \in \mathbb{R}^{N-1}$. The interconnection matrix satisfies the skew-symmetric property $\boldsymbol{J}(\boldsymbol{x}) = -\boldsymbol{J}^T(\boldsymbol{x})$ which is its power conserving property. The resistive structure in $\boldsymbol{R}(\boldsymbol{x})$ satisfies $\boldsymbol{R}(\boldsymbol{x}) = \boldsymbol{R}^T(\boldsymbol{x}) \geq 0$ which represents the energy dissipation (and thus stability) of the system.

To derive the actual vectors and matrices $\boldsymbol{J}(\boldsymbol{x})$,$\boldsymbol{R}(\boldsymbol{x})$ and $\boldsymbol{g}(\boldsymbol{x})$ the following equation manipulations are necessary. Combining the equations from 4.10 with the equations from 4.13 and using equation 5.1 for $\boldsymbol{u}$ gives the port-Hamiltonian input-state-output representation from equation A.10.

$$\begin{bmatrix}\dot{\boldsymbol{q}}\\ \dot{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\mathbf{0} & \boldsymbol{I}\\ -\boldsymbol{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\frac{\partial H}{\partial \boldsymbol{q}}\\ \frac{\partial H}{\partial \boldsymbol{p}}\end{bmatrix} + \begin{bmatrix}\mathbf{0}\\ \boldsymbol{U_{ext}}\end{bmatrix} \tag{A.10}$$

To get to the desired shape from equation 4.13 (necessary for controller design) $\boldsymbol{J}(\boldsymbol{x})$ is straight forward and becomes $\begin{bmatrix}\mathbf{0} & \boldsymbol{I}\\ -\boldsymbol{I} & \mathbf{0}\end{bmatrix}$. $\boldsymbol{R}(\boldsymbol{x})$ is determined via rewriting the equations from 4.4 to:

$$\boldsymbol{R}(\boldsymbol{x}) = \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{R_f}\end{bmatrix} \tag{A.11a}$$

$$\boldsymbol{R_f} = -\begin{bmatrix}l\boldsymbol{S_\theta}\mathbf{K} & -l\boldsymbol{C_\theta}\boldsymbol{K}\\ \boldsymbol{I} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}c_t(\boldsymbol{C_\theta})^2 + c_n(\boldsymbol{S_\theta})^2 & (c_t - c_n)\boldsymbol{S_\theta}\boldsymbol{C_\theta}\\ (c_t - c_n)\boldsymbol{S_\theta}\boldsymbol{C_\theta} & c_t(\boldsymbol{S_\theta})^2 + c_n(\boldsymbol{C_\theta})^2\end{bmatrix}\begin{bmatrix}l\boldsymbol{K}^T\boldsymbol{S_\theta} & \boldsymbol{I}^T & \mathbf{0}^T\\ -l\boldsymbol{K}^T\boldsymbol{C_\theta} & \mathbf{0}^T & \mathbf{I}^T\end{bmatrix} \tag{A.11b}$$

$\boldsymbol{S_\theta}$,$\boldsymbol{C_\theta}$,$\boldsymbol{K} \in \mathbb{R}^{NXN}$ and $\boldsymbol{I}$,$\mathbf{0} \in \mathbb{R}^{1XN}$.

The resulting $\boldsymbol{g(x)}$ becomes:

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} \mathbf{0}_1 \\ \boldsymbol{D}^T \\ \mathbf{0}_2 \end{bmatrix} \tag{A.12}$$

Where $\mathbf{0}_1 \in \mathbb{R}^{(N+2)X(N-1)}$ and $\mathbf{0}_2 \in \mathbb{R}^{2X(N-1)}$

A.9 Newton-Euler controller design: symbol definitions

The equations for the variables $\bar{\boldsymbol{M}}$,$\bar{\boldsymbol{W}}$,$\bar{\boldsymbol{G}}$ and $\bar{\boldsymbol{B}}$ are:

$$\bar{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{H}^T\boldsymbol{M}_\theta(\bar{\phi})\boldsymbol{H} & \mathbf{0}_{N\times 2} \\ \mathbf{0}_{N\times 2} & Nm\mathbf{I}_2 \end{bmatrix} \tag{A.13a}$$

$$\bar{\boldsymbol{W}} = \begin{bmatrix} \boldsymbol{H}^T\boldsymbol{W}(\bar{\phi})diag(\mathbf{H}\dot{\bar{\boldsymbol{\phi}}})\mathbf{H}\dot{\bar{\boldsymbol{\phi}}} \\ \mathbf{0}_{2\times 1} \end{bmatrix} \tag{A.13b}$$

$$\bar{\boldsymbol{G}} = \begin{bmatrix} -l\boldsymbol{H}^T\boldsymbol{S}_\theta\mathbf{K} & l\boldsymbol{H}^T\boldsymbol{C}_\theta\mathbf{K} \\ -\mathbf{e}^T & \mathbf{0}_{1\times N} \\ \mathbf{0}_{1\times N} & -\mathbf{e}^T \end{bmatrix} \tag{A.13c}$$

$$\bar{\boldsymbol{B}} = \begin{bmatrix} \mathbf{I}_{N-1} \\ \mathbf{0}_{3\times(N-1)} \end{bmatrix} \tag{A.13d}$$

A.10 Newton-Euler controller design: controller equations

The simple joint controller is given by:

$$\bar{u} = k_p(\phi_{ref} - \phi) - k_d\dot{\phi} \tag{A.14}$$

where $k_p > 0$ and $k_d > 0$

The exponentially stable joint controller is given by:

$$\bar{u} = \ddot{\phi}_{ref} + k_d(\dot{\phi}_{ref} - \dot{\phi}) + k_p(\phi_{ref} - \phi) \tag{A.15}$$

where $k_p > 0$ and $k_d > 0$

A.11 port-Hamiltonian Input-State-Output coordinate transformation

The port-Hamiltonian input-state-output representation for the 4 links snake robot in actuated and unactuated coordinates is given in equation A.16:

$$\begin{bmatrix} \dot{\bar{\phi}} \\ \dot{p}_x \\ \dot{p}_y \\ \dot{\bar{\phi}}_p \\ \dot{p}_5 \\ \dot{p}_6 \end{bmatrix} = \begin{bmatrix} (H^TH)^{-1}(H^T((J(x)(1:4,:) - R(x)(1:4,:)) \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix}) + H^Tg(x)(1:4,:) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}) \\ (J(x)(5:6,:) - R(x)(5:6,:)) \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + g(x)(5:6,:) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \\ (H^TH)^{-1}(H^T((J(x)(7:10,:) - R(x)(7:10,:)) \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix}) + H^Tg(x)(7:10,:) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}) \\ (J(x)(11:12,:) - R(x)(11:12,:)) \begin{bmatrix} \frac{\partial H}{\partial q} \\ \frac{\partial H}{\partial p} \end{bmatrix} + g(x)(11:12,:) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \end{bmatrix}$$

$\dot{\bar{\phi}} = [\dot{\phi} \quad \dot{\theta}_4]^T$ and $\dot{\bar{\phi}}_p = [\dot{\phi}_p \quad \dot{p}_4]^T$.

Combining this port-Hamiltonian Input-State-Output matrix with equation A.12 it becomes clear that the actuator torques $\boldsymbol{u}$ only enter the system via $\dot{\phi}_p$. From also combining equation A.11a with the matrix it becomes clear that the friction forces on the hinges also enter the system via $\dot{\phi}_p$.

The result of $(H^TH)^{-1}H^Tg(x)(7:10,:)u$:

$$(H^TH)^{-1}H^Tg(x)(7:10,:) \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 2u_1 - u_2 \\ 2u_2 - u_1 - u_3 \\ 2u_3 - u_2 \end{bmatrix} \tag{A.17}$$

Which is used to check the result of the coordinate transformation. It is done correctly since $\phi_1 = \theta_1 - \theta_2$ and θ_1 is influenced by u_1 from actuator 1. θ_2 is influenced by u_1 (from actuator 1) and u_2 (from actuator 2) so ϕ_1 is influenced by u_1 twice and once by u_2. Comparable arguments prove the same for ϕ_2 and ϕ_3.

A.12 $\frac{dC_i}{dt} = 0$ for the second and third Casimir

For the second Casimir (equation 4.20b) $\frac{dC_i}{dt} = 0$ is:

$$\frac{dC_i}{dt} = \begin{bmatrix} (a_1(\phi_{q,1,ref} - \phi_{q,1}) + b_1(\phi_{p,1,ref} - \phi_{p,1}))(a_1(\dot{\phi}_{q,1,ref} - \dot{\phi}_{q,1}) + b_1(\dot{\phi}_{p,1,ref} - \dot{\phi}_{p,1})) \\ (a_2(\phi_{q,2,ref} - \phi_{q,2}) + b_2(\phi_{p,2,ref} - \phi_{p,2}))(a_2(\dot{\phi}_{q,2,ref} - \dot{\phi}_{q,2}) + b_2(\dot{\phi}_{p,2,ref} - \dot{\phi}_{p,2})) \\ (a_3(\phi_{q,3,ref} - \phi_{q,3}) + b_3(\phi_{p,3,ref} - \phi_{p,3}))(a_3(\dot{\phi}_{q,3,ref} - \dot{\phi}_{q,3}) + b_3(\dot{\phi}_{p,3,ref} - \dot{\phi}_{p,3})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.18}$$

For the third Casimir (equation 4.20c) $\frac{dC_i}{dt} = 0$ is:

$$\frac{dC_i}{dt} = \begin{bmatrix} (a_1(\phi_{p,1,ref} - \phi_{p,1})(\dot{\phi}_{q,1,ref} - \dot{\phi}_{q,1})) + (a_1(\phi_{q,1,ref} - \phi_{q,1})(\dot{\phi}_{p,1,ref} - \dot{\phi}_{p,1})) \\ (a_2(\phi_{p,2,ref} - \phi_{p,2})(\dot{\phi}_{q,2,ref} - \dot{\phi}_{q,2})) + (a_2(\phi_{q,2,ref} - \phi_{q,2})(\dot{\phi}_{p,2,ref} - \dot{\phi}_{p,2})) \\ (a_3(\phi_{p,3,ref} - \phi_{p,3})(\dot{\phi}_{q,3,ref} - \dot{\phi}_{q,3})) + (a_3(\phi_{q,3,ref} - \phi_{q,3})(\dot{\phi}_{p,3,ref} - \dot{\phi}_{p,3})) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.19}$$

A.13 Energy-Casimir method: control laws

Control law for u_1 obtained from solving 4.21 corresponding to the first Casimir (equation 4.20a) (the control laws for u_2 and u_3 are very comparable):

$\mathrm{u}_1 =$

$$-(3\phi_{q,1}\phi_{p,2}\phi_{p,3}a_1b_2b_3\dot\phi_{q,1} + 2\phi_{q,2}\phi_{p,1}\phi_{p,3}a_2b_1b_3\dot\phi_{q,2} + \phi_{q,3}\phi_{p,1}\phi_{p,2}a_3b_1b_2\dot\phi_{q,3} - 3\phi_{q,1}\phi_{p,2}\phi_{p,3}a_1b_2b_3\dot\phi_{q,1,ref} -$$
$$3\phi_{q,1}\phi_{p,2}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1} - 3\phi_{q,1}\phi_{p,3}\phi_{p,2,ref}a_1b_2b_3\dot\phi_{q,1} - 3\phi_{p,2}\phi_{p,3}\phi_{q,1,ref}a_1b_2b_3\dot\phi_{q,1} - 2\phi_{q,2}\phi_{p,1}\phi_{p,3}a_2b_1b_3\dot\phi_{q,2,ref} -$$
$$2\phi_{q,2}\phi_{p,1}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2} - 2\phi_{q,2}\phi_{p,3}\phi_{p,1,ref}a_2b_1b_3\dot\phi_{q,2} - 2\phi_{p,1}\phi_{p,3}\phi_{q,2,ref}a_2b_1b_3\dot\phi_{q,2} - \phi_{q,3}\phi_{p,1}\phi_{p,2}a_3b_1b_2\dot\phi_{q,3,ref} -$$
$$\phi_{q,3}\phi_{p,1}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3} - \phi_{q,3}\phi_{p,2}\phi_{p,1,ref}a_3b_1b_2\dot\phi_{q,3} - \phi_{p,1}\phi_{p,2}\phi_{q,3,ref}a_3b_1b_2\dot\phi_{q,3} + 3\phi_{q,1}\phi_{p,2}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1,ref} +$$
$$3\phi_{q,1}\phi_{p,3}\phi_{p,2,ref}a_1b_2b_3\dot\phi_{q,1,ref} + 3\phi_{q,1}\phi_{p,2,ref}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1} + 3\phi_{p,2}\phi_{p,3}\phi_{q,1,ref}a_1b_2b_3\dot\phi_{q,1,ref} + 3\phi_{p,2}\phi_{q,1,ref}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1}$$
$$3\phi_{p,3}\phi_{q,1,ref}\phi_{p,2,ref}a_1b_2b_3\dot\phi_{q,1} + 2\phi_{q,2}\phi_{p,1}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2,ref} + 2\phi_{q,2}\phi_{p,3}\phi_{p,1,ref}a_2b_1b_3\dot\phi_{q,2,ref} + 2\phi_{q,2}\phi_{p,1,ref}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2}$$
$$2\phi_{p,1}\phi_{p,3}\phi_{q,2,ref}a_2b_1b_3\dot\phi_{q,2,ref} + 2\phi_{p,1}\phi_{q,2,ref}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2} + 2\phi_{p,3}\phi_{q,2,ref}\phi_{p,1,ref}a_2b_1b_3\dot\phi_{q,2} + \phi_{q,3}\phi_{p,1}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3,ref} +$$
$$\phi_{q,3}\phi_{p,2}\phi_{p,1,ref}a_3b_1b_2\dot\phi_{q,3,ref} + \phi_{q,3}\phi_{p,1,ref}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3} + \phi_{p,1}\phi_{p,2}\phi_{q,3,ref}a_3b_1b_2\dot\phi_{q,3,ref} + \phi_{p,1}\phi_{q,3,ref}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3} +$$
$$\phi_{p,2}\phi_{q,3,ref}\phi_{p,1,ref}a_3b_1b_2\dot\phi_{q,3} - 3\phi_{q,1}\phi_{p,2,ref}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1,ref} - 3\phi_{p,2}\phi_{q,1,ref}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1,ref} - 3\phi_{p,3}\phi_{q,1,ref}\phi_{p,2,ref}a_1b_2$$
$$3\phi_{q,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1} - 2\phi_{q,2}\phi_{p,1,ref}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2,ref} - 2\phi_{p,1}\phi_{q,2,ref}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2,ref} -$$
$$2\phi_{p,3}\phi_{q,2,ref}\phi_{p,1,ref}a_2b_1b_3\dot\phi_{q,2,ref} - 2\phi_{q,2,ref}\phi_{p,1,ref}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2} - \phi_{q,3}\phi_{p,1,ref}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3,ref} - \phi_{p,1}\phi_{q,3,ref}\phi_{p,2,ref}a_3$$
$$\phi_{p,2}\phi_{q,3,ref}\phi_{p,1,ref}a_3b_1b_2\dot\phi_{q,3,ref} - \phi_{q,3,ref}\phi_{p,1,ref}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3} + 3\phi_{q,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}a_1b_2b_3\dot\phi_{q,1,ref} +$$
$$2\phi_{q,2,ref}\phi_{p,1,ref}\phi_{p,3,ref}a_2b_1b_3\dot\phi_{q,2,ref} + \phi_{q,3,ref}\phi_{p,1,ref}\phi_{p,2,ref}a_3b_1b_2\dot\phi_{q,3,ref} + 3\phi_{p,1}\phi_{p,2}\phi_{p,3}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} + 2\phi_{p,1}\phi_{p,2}\phi_{p,3}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} +$$
$$\phi_{p,1}\phi_{p,2}\phi_{p,3}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} - 3\phi_{p,1}\phi_{p,2}\phi_{p,3}b_1b_2b_3\dot\phi_{p,1,ref} - 2\phi_{p,1}\phi_{p,2}\phi_{p,3}b_1b_2b_3\dot\phi_{p,2,ref} - \phi_{p,1}\phi_{p,2}\phi_{p,3}b_1b_2b_3\dot\phi_{p,3,ref} -$$
$$3\phi_{p,1}\phi_{p,2}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} - 3\phi_{p,1}\phi_{p,3}\phi_{p,2,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} - 3\phi_{p,2}\phi_{p,3}\phi_{p,1,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} - 2\phi_{p,1}\phi_{p,2}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} -$$
$$2\phi_{p,1}\phi_{p,3}\phi_{p,2,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} - 2\phi_{p,2}\phi_{p,3}\phi_{p,1,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} - \phi_{p,1}\phi_{p,2}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} - \phi_{p,1}\phi_{p,3}\phi_{p,2,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} -$$
$$\phi_{p,2}\phi_{p,3}\phi_{p,1,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} + 3\phi_{p,1}\phi_{p,2}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,1,ref} + 3\phi_{p,1}\phi_{p,3}\phi_{p,2,ref}b_1b_2b_3\dot\phi_{p,1,ref} + 3\phi_{p,2}\phi_{p,3}\phi_{p,1,ref}b_1b_2b_3\dot\phi_{p,1,ref} +$$
$$2\phi_{p,1}\phi_{p,2}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,2,ref} + 2\phi_{p,1}\phi_{p,3}\phi_{p,2,ref}b_1b_2b_3\dot\phi_{p,2,ref} + 2\phi_{p,2}\phi_{p,3}\phi_{p,1,ref}b_1b_2b_3\dot\phi_{p,2,ref} + \phi_{p,1}\phi_{p,2}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,3,ref} +$$
$$\phi_{p,1}\phi_{p,3}\phi_{p,2,ref}b_1b_2b_3\dot\phi_{p,3,ref} + \phi_{p,2}\phi_{p,3}\phi_{p,1,ref}b_1b_2b_3\dot\phi_{p,3,ref} + 3\phi_{p,1}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} + 3\phi_{p,2}\phi_{p,1,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} +$$
$$3\phi_{p,3}\phi_{p,1,ref}\phi_{p,2,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} + 2\phi_{p,1}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} + 2\phi_{p,2}\phi_{p,1,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} + 2\phi_{p,3}\phi_{p,1,ref}\phi_{p,2,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2}$$
$$\phi_{p,1}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} + \phi_{p,2}\phi_{p,1,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} + \phi_{p,3}\phi_{p,1,ref}\phi_{p,2,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,3} - 3\phi_{p,1}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,1,ref}$$
$$3\phi_{p,2}\phi_{p,1,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,1,ref} - 3\phi_{p,3}\phi_{p,1,ref}\phi_{p,2,ref}b_1b_2b_3\dot\phi_{p,1,ref} - 2\phi_{p,1}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,2,ref} -$$
$$2\phi_{p,2}\phi_{p,1,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,2,ref} - 2\phi_{p,3}\phi_{p,1,ref}\phi_{p,2,ref}b_1b_2b_3\dot\phi_{p,2,ref} - \phi_{p,1}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,3,ref} - \phi_{p,2}\phi_{p,1,ref}\phi_{p,3,ref}b_1b$$
$$\phi_{p,3}\phi_{p,1,ref}\phi_{p,2,ref}b_1b_2b_3\dot\phi_{p,3,ref} - 3\phi_{p,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,1} - 2\phi_{p,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot{\tilde{\phi}}_{p,2} - \phi_{p,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}$$
$$3\phi_{p,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,1,ref} + 2\phi_{p,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,2,ref} + \phi_{p,1,ref}\phi_{p,2,ref}\phi_{p,3,ref}b_1b_2b_3\dot\phi_{p,3,ref})/(4b_1b_2b_3(\phi_p$$
$$\phi_{p,1,ref})(\phi_{p,2} - \phi_{p,2,ref})(\phi_{p,3} - \phi_{p,3,ref}))$$

In the equations for u_1 obtained from the first and third Casimir there is the term $\dot{\tilde{\phi}}_{p,i}$. As explained in appendix A.11 $\dot\phi_{p,i}$ contains both the actuator torques $\boldsymbol{u}$ and the friction forces on the hinges. When the systems of equations resulting from $\frac{dC_i}{dt}$ are solved for $\boldsymbol{u}$ the term $\dot\phi_{p,i}$ is split. The resulting term is denoted as $\dot{\tilde{\phi}}_{p,i}$. Therefore $\dot{\tilde{\phi}}_{p,i}$ is $\dot\phi_{p,i}$ without the actuator torques.

Control laws for u_1,u_2 and u_3 obtained from solving A.18 corresponding to the second Casimir (equation 4.20b):

$$u_1 = -(3a_1b_2b_3\dot\phi_{q,1} + 2a_2b_1b_3\dot\phi_{q,2} + a_3b_1b_2\dot\phi_{q,3} - 3a_1b_2b_3\dot\phi_{q,1,ref} - 2a_2b_1b_3\dot\phi_{q,2,ref} - a_3b_1b_2\dot\phi_{q,3,ref} + 3b_1b_2b_3\dot{\tilde{\phi}}_{p,1} + 2b_1b_2b_3\dot{\tilde{\phi}}_{p,2} + b_1b_2b_3\dot{\tilde{\phi}}_{p,3} - 3b_1b_2b_3\dot\phi_{p,1,ref} - 2b_1b_2b_3\dot\phi_{p,2,ref} - b_1b_2b_3\dot\phi_{p,3,ref})/(4b_1b_2b_3) \tag{A.20}$$

$$u_2 = -(a_1b_2b_3\dot\phi_{q,1} + 2a_2b_1b_3\dot\phi_{q,2} + a_3b_1b_2\dot\phi_{q,3} - a_1b_2b_3\dot\phi_{q,1,ref} - 2a_2b_1b_3\dot\phi_{q,2,ref} - a_3b_1b_2\dot\phi_{q,3,ref} + b_1b_2b_3\dot{\tilde{\phi}}_{p,1} + 2b_1b_2b_3\dot{\tilde{\phi}}_{p,2} + b_1b_2b_3\dot{\tilde{\phi}}_{p,3} - b_1b_2b_3\dot\phi_{p,1,ref} - 2b_1b_2b_3\dot\phi_{p,2,ref} - b_1b_2b_3\dot\phi_{p,3,ref})/(2b_1b_2b_3) \tag{A.21}$$

$$u_3 = (a_1b_2b_3\dot\phi_{q,1} + 2a_2b_1b_3\dot\phi_{q,2} + 3a_3b_1b_2\dot\phi_{q,3} - a_1b_2b_3\dot\phi_{q,1,ref} - 2a_2b_1b_3\dot\phi_{q,2,ref} - 3a_3b_1b_2\dot\phi_{q,3,ref} + b_1b_2b_3\dot{\tilde{\phi}}_{p,1} + 2b_1b_2b_3\dot{\tilde{\phi}}_{p,2} + 3b_1b_2b_3\dot{\tilde{\phi}}_{p,3} - b_1b_2b_3\dot\phi_{p,1,ref} - 2b_1b_2b_3\dot\phi_{p,2,ref} - 3b_1b_2b_3\dot\phi_{p,3,ref})/(4b_1b_2b_3) \tag{A.22}$$

Control law for u_1 obtained from solving A.19 corresponding to the third Casimir (equation 4.20c) (the control laws for u_2 and u_3 are very comparable):

$\mathrm{u}_1 =$

$$-(3\phi_{q,2}\phi_{q,3}\phi_{p,1}\dot{\phi}_{q,1}+2\phi_{q,1}\phi_{q,3}\phi_{p,2}\dot{\phi}_{q,2}+\phi_{q,1}\phi_{q,2}\phi_{p,3}\dot{\phi}_{q,3}+3\phi_{q,1}\phi_{q,2}\phi_{q,3}\dot{\tilde{\phi}}_{p,1}+2\phi_{q,1}\phi_{q,2}\phi_{q,3}\dot{\tilde{\phi}}_{p,2}+\phi_{q,1}\phi_{q,2}\phi_{q,3}\dot{\tilde{\phi}}_{p,3}-$$
$$3\phi_{q,1}\phi_{q,2}\phi_{q,3}\dot{\phi}_{p,1,ref}-3\phi_{q,2}\phi_{q,3}\phi_{p,1}\dot{\phi}_{q,1,ref}-3\phi_{q,2}\phi_{q,3}\phi_{p,1,ref}\dot{\phi}_{q,1}-3\phi_{q,2}\phi_{p,1}\phi_{q,3,ref}\dot{\phi}_{q,1}-3\phi_{q,3}\phi_{p,1}\phi_{q,2,ref}\dot{\phi}_{q,1}-$$
$$2\phi_{q,1}\phi_{q,2}\phi_{q,3}\dot{\phi}_{p,2,ref}-2\phi_{q,1}\phi_{q,3}\phi_{p,2}\dot{\phi}_{q,2,ref}-2\phi_{q,1}\phi_{q,3}\phi_{p,2,ref}\dot{\phi}_{q,2}-2\phi_{q,1}\phi_{p,2}\phi_{q,3,ref}\dot{\phi}_{q,2}-2\phi_{q,3}\phi_{p,2}\phi_{q,1,ref}\dot{\phi}_{q,2}-$$
$$\phi_{q,1}\phi_{q,2}\phi_{q,3}\dot{\phi}_{p,3,ref} - \phi_{q,1}\phi_{q,2}\phi_{p,3}\dot{\phi}_{q,3,ref} - \phi_{q,1}\phi_{q,2}\phi_{p,3,ref}\dot{\phi}_{q,3} - \phi_{q,1}\phi_{p,3}\phi_{q,2,ref}\dot{\phi}_{q,3} - \phi_{q,2}\phi_{p,3}\phi_{q,1,ref}\dot{\phi}_{q,3} -$$
$$3\phi_{q,1}\phi_{q,2}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,1}-3\phi_{q,1}\phi_{q,3}\phi_{q,2,ref}\dot{\tilde{\phi}}_{p,1}-3\phi_{q,2}\phi_{q,3}\phi_{q,1,ref}\dot{\tilde{\phi}}_{p,1}-2\phi_{q,1}\phi_{q,2}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,2}-2\phi_{q,1}\phi_{q,3}\phi_{q,2,ref}\dot{\tilde{\phi}}_{p,2}-$$
$$2\phi_{q,2}\phi_{q,3}\phi_{q,1,ref}\dot{\tilde{\phi}}_{p,2}-\phi_{q,1}\phi_{q,2}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,3}-\phi_{q,1}\phi_{q,3}\phi_{q,2,ref}\dot{\tilde{\phi}}_{p,3}-\phi_{q,2}\phi_{q,3}\phi_{q,1,ref}\dot{\tilde{\phi}}_{p,3}+3\phi_{q,1}\phi_{q,2}\phi_{q,3,ref}\dot{\phi}_{p,1,ref}+$$
$$3\phi_{q,1}\phi_{q,3}\phi_{q,2,ref}\dot{\phi}_{p,1,ref}+3\phi_{q,2}\phi_{q,3}\phi_{q,1,ref}\dot{\phi}_{p,1,ref}+3\phi_{q,2}\phi_{q,3}\phi_{p,1,ref}\dot{\phi}_{q,1,ref}+3\phi_{q,2}\phi_{p,1}\phi_{q,3,ref}\dot{\phi}_{q,1,ref}+3\phi_{q,2}\phi_{q,3,ref}\phi_{p,1,ref}\dot{\phi}_{q,1}$$
$$3\phi_{q,3}\phi_{p,1}\phi_{q,2,ref}\dot{\phi}_{q,1,ref}+3\phi_{q,3}\phi_{q,2,ref}\phi_{p,1,ref}\dot{\phi}_{q,1}+3\phi_{p,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{q,1}+2\phi_{q,1}\phi_{q,2}\phi_{q,3,ref}\dot{\phi}_{p,2,ref}+2\phi_{q,1}\phi_{q,3}\phi_{q,2,ref}\dot{\phi}_{p,2,ref}$$
$$2\phi_{q,1}\phi_{q,3}\phi_{p,2,ref}\dot{\phi}_{q,2,ref}+2\phi_{q,1}\phi_{p,2}\phi_{q,3,ref}\dot{\phi}_{q,2,ref}+2\phi_{q,1}\phi_{q,3,ref}\phi_{p,2,ref}\dot{\phi}_{q,2}+2\phi_{q,2}\phi_{q,3}\phi_{q,1,ref}\dot{\phi}_{p,2,ref}+2\phi_{q,3}\phi_{p,2}\phi_{q,1,ref}\dot{\phi}_{q,2,ref}$$
$$2\phi_{q,3}\phi_{q,1,ref}\phi_{p,2,ref}\dot{\phi}_{q,2}+2\phi_{p,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\phi}_{q,2}+\phi_{q,1}\phi_{q,2}\phi_{q,3,ref}\dot{\phi}_{p,3,ref}+\phi_{q,1}\phi_{q,2}\phi_{p,3,ref}\dot{\phi}_{q,3,ref}+\phi_{q,1}\phi_{q,3}\phi_{q,2,ref}\dot{\phi}_{p,3,ref}+$$
$$\phi_{q,1}\phi_{p,3}\phi_{q,2,ref}\dot{\phi}_{q,3,ref}+\phi_{q,1}\phi_{q,2,ref}\phi_{p,3,ref}\dot{\phi}_{q,3}+\phi_{q,2}\phi_{q,3}\phi_{q,1,ref}\dot{\phi}_{p,3,ref}+\phi_{q,2}\phi_{p,3}\phi_{q,1,ref}\dot{\phi}_{q,3,ref}+\phi_{q,2}\phi_{q,1,ref}\phi_{p,3,ref}\dot{\phi}_{q,3}+$$
$$\phi_{p,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\phi}_{q,3}+3\phi_{q,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,1}+3\phi_{q,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,1}+3\phi_{q,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\tilde{\phi}}_{p,1}+2\phi_{q,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,2}+$$
$$2\phi_{q,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,2}+2\phi_{q,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\tilde{\phi}}_{p,2}+\phi_{q,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,3}+\phi_{q,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,3}+\phi_{q,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\tilde{\phi}}_{p,3}-$$
$$3\phi_{q,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{p,1,ref}-3\phi_{q,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\phi}_{p,1,ref}-3\phi_{q,2}\phi_{q,3,ref}\phi_{p,1,ref}\dot{\phi}_{q,1,ref}-3\phi_{q,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\phi}_{p,1,ref}-$$
$$3\phi_{q,3}\phi_{q,2,ref}\phi_{p,1,ref}\dot{\phi}_{q,1,ref}-3\phi_{p,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{q,1,ref}-3\phi_{q,2,ref}\phi_{q,3,ref}\phi_{p,1,ref}\dot{\phi}_{q,1}-2\phi_{q,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{p,2,ref}-$$
$$2\phi_{q,1}\phi_{q,3,ref}\phi_{p,2,ref}\dot{\phi}_{q,2,ref}-2\phi_{q,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\phi}_{p,2,ref}-2\phi_{q,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\phi}_{p,2,ref}-2\phi_{q,3}\phi_{q,1,ref}\phi_{p,2,ref}\dot{\phi}_{q,2,ref}-$$
$$2\phi_{p,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\phi}_{q,2,ref}-2\phi_{q,1,ref}\phi_{q,3,ref}\phi_{p,2,ref}\dot{\phi}_{q,2}-\phi_{q,1}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{p,3,ref}-\phi_{q,1}\phi_{q,2,ref}\phi_{p,3,ref}\dot{\phi}_{q,3,ref}-$$
$$\phi_{q,2}\phi_{q,1,ref}\phi_{q,3,ref}\dot{\phi}_{p,3,ref}-\phi_{q,2}\phi_{q,1,ref}\phi_{p,3,ref}\dot{\phi}_{q,3,ref}-\phi_{q,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\phi}_{p,3,ref}-\phi_{p,3}\phi_{q,1,ref}\phi_{q,2,ref}\dot{\phi}_{q,3,ref}-$$
$$\phi_{q,1,ref}\phi_{q,2,ref}\phi_{p,3,ref}\dot{\phi}_{q,3}-3\phi_{q,1,ref}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,1}-2\phi_{q,1,ref}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,2}-\phi_{q,1,ref}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\tilde{\phi}}_{p,3}+$$
$$3\phi_{q,1,ref}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{p,1,ref}+3\phi_{q,2,ref}\phi_{q,3,ref}\phi_{p,1,ref}\dot{\phi}_{q,1,ref}+2\phi_{q,1,ref}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{p,2,ref}+2\phi_{q,1,ref}\phi_{q,3,ref}\phi_{p,2,ref}\dot{\phi}_{q,2,re}$$
$$\phi_{q,1,ref}\phi_{q,2,ref}\phi_{q,3,ref}\dot{\phi}_{p,3,ref}+\phi_{q,1,ref}\phi_{q,2,ref}\phi_{p,3,ref}\dot{\phi}_{q,3,ref})/(4(\phi_{q,1}-\phi_{q,1,ref})(\phi_{q,2}-\phi_{q,2,ref})(\phi_{q,3}-\phi_{q,3,ref}))$$

B Reflection report

B.0.1 Introduction

Nowadays the world is getting smaller and more focus is on international and multicultural collaborations. Our university is promoting internationalisation in numerous ways. One of them is changing the language of our mechanical engineering bachelor to English. Seeing our university becoming more and more international made me realize that I myself was not becoming very internationally oriented. I realized that almost all my friends where white Dutch guys and (a few) girls. Furthermore I was speaking Dutch in my student home and for group assignments in my master I was always working with fellow Dutch white students. I realized I was missing out on the internationalization and that is the reason I decided to go abroad for my internship. Preferably far away and definitely outside of Europe.

In this report I formulate my SMART learning objectives in the Learning part of the STARL-method to prevent writing the same things twice. Furthermore I refer to my graduation assignment a few times. To give this more context, I am going to do my assignment at the (multicultural) company ASML.

B.0.2 Professional learning objectives

For the STARL-method I will refer to the following 2 situations/tasks to reflect on my professional learning goals:

Situation 1: I had a personal progress meeting with my supervisor once a week in her office. My supervisor is an associate professor originally from Malaysia. She is way more introverted and relaxed than me. This may be related to the culture she is from. Furthermore she is living and working in Adelaide Australia for a few years now.
Task 1: The goals of these personal progress meetings where to discuss my progress, to determine future directions of the research and to ask questions related to my research.

Situation 2: I had a project progress meeting with 4 master students (who are originally from China) and my supervisor once a week. These Chinese students did not speak up and where very passive in their communication. This may be related to the culture they are raised in.
Task 2: The goal of this meeting was to discuss the progress of the whole project. The Chinese students had the task to design and build a physical snake robot. My task was to model the snake robot dynamics and to design a controller for the robot. The long term goal was to combine the work and test my controller on their physical snake robot.

Professional learning objective 1: I know that Dutch people can be very direct and say what they think. In the Netherlands we think that it is a good character trait and I have the feeling that we are not always aware of the fact that other people can think of us a rude. When I look at myself I know that I can be direct and I am not afraid to speak up when I disagree with someone. Furthermore when an important idea pops up in my head I sometimes have the bad habit to speak it out loud directly and thus interrupt people while they are talking. I am aware that I am doing that and I am already paying attention to it the Netherlands. In international teamwork it is even more important for me to be patient and let other people finish their story first. Therefore my first personal development goal is:

Be patient in team meetings. Give other people enough room to speak out loud and actively ask other people for input during the meetings.

Result/reflect: My supervisor asked questions during the team meetings (situation/task 2). I actively suppressed my urge to directly answer here questions and gave the Chinese students time to think. This went well most of the time although it also happened a few times that I lost my patients and just answered the questions after seeing that the Chinese students again did not know where we were talking about. In these group meetings I also had to explain certain snake dynamics and kinematics to the team and my supervisor. The Chinese students did not ask any questions so when I could see from their faces that they did not understand it I actively asked them if they understood it and where I could clarify certain stuff. However also here at certain moments I lost my patience and just continued my story once I saw my supervisor understood me (even though I could see that the Chinese students again did not understand what I was telling). During the personal meetings with my supervisor (situation/task 1) I sometimes forgot to verify enough if my supervisor understood my story/ followed my analysis. Most of the times if that happened I realised that after our meetings. The moments I felt I did not check that enough I wrote it down and made sure I payed more attention to it our next meeting.

Learning: From the team meetings I really learned that people from (some) other cultures do not dare to speak up. I already read about this prior to going to Australia but now I experienced it. Although I actively tried to take that into consideration I did lose my patience and willingness to take it into consideration a few times. It was a very valuable experience. I think the interaction and teamwork with my supervisor went very well. I really learned to give her enough space during our meetings.
SMART learning objective: I will work on being more patient in (multicultural) group meetings during my graduation assignment. To measure this I will ask my fellow team members how they experience me during our group meetings.

Professional learning objective 2: English is not my native language which makes proper communication in a multinational team environment more challenging. My English proficiency is adequate to communicate properly on a daily basis. However I could definitely improve. From this follows my second personal development goal is:

Communicate in English about complex technical topics with colleagues.

Result/reflect: With my Malaysian supervisor the communication went well both in our personal meetings (situation/task 1) as during the team meetings (situation/task 2). This was due to the high level of English proficiency of my supervisor. She took the time for our meetings so there was enough room for me to explain what I wanted to tell, even though I did not always knew the specific words for points I wanted to convey. It was a different story when we had meetings with my Chinese teammates. My Chinese teammates where not well acquainted with English. Especially when I explained certain complex dynamics of the snake robot I could see that they often where not able to understand my explanation. During meetings I sometimes had to explain the same thing three times before I had the feeling they kind of understood what I was trying to say. On the other hand the English articulation of my Chinese teammates was very bad. This made understanding them sometimes very very difficult. After a few meetings I decided to summarize what I think they wanted to explain to me to check if I understood them correctly. This helped a lot and the communication between us drastically improved.

Learning: The main thing I got out of this learning goal is that for effective communication it is very helpful to summarize what you think is the conclusion of your discussion and verify that with the other party. Further more the lack of English proficiency of my fellow students motivates me to improve my English vocabulary. This way if people do not understand my message I can use my extended vocabulary to convey the same message in different words.
SMART learning objective: In (multicultural) group meetings during my graduation I will (when I think it is suitable) summarize discussions and verify it with the other parties involved.
SMART learning objective: I am going to read at least 3 English books before September 2020 to improve my English vocabulary.

B.0.3 Personal learning objectives

For the STARL-method I will refer to the following 3 situations/tasks to reflect on my personal learning goals:

Situation 3: During my time in Adelaide I stayed at a student college with approximately 25 other (PhD) students (and a few lecturers) of the university of Adelaide and university of South-Australia. Everybody had his or her own room. We shared kitchen, bathroom and 2 big community living rooms together.
Task 3: My (personal) ”task” was to make friends/ building a new social network.

Situation 4: My friend had a group of friends where she hang out with every Friday night. When my friend was still at the college she asked me twice to join them. I was glad I could join since otherwise I had nothing to do.
Task 4: Again the "task" was to make new friends

Situation 5: Most of my time I spend working behind my computer in my office. The office was a room without daylight and it always smelled musty. I had one Dutch student sitting next to me. Further more there where two French students working there. Unfortunately we could not communicate with them because they did not speak English. Also there where a few research students from the University of Adelaide working in our office irregularly.
Task 5: My main tasks where to model the snake robot dynamics and design a control law. In practice I was sitting behind my desk the whole day reading literature, programming/ debugging my code and doing analyses and derivations on paper.

Personal learning objective 1: Besides developing myself in international teamwork and communication I also had some personal developing goals. While studying for more than 5 years in the Enschede I developed a broad social network. I knew a lot of people in Enschede and I had plenty of friends to hang out with. This was very nice but it also made me a little bit lazy. I did not meet a lot of new people because there was not a lot of incentive for me to do so. After those 5 years I thought it was time for something new (also related to my social life). I wanted to start again and meet new people so my first personal learning goal was:

Start from scratch with building a new social life (in Australia).

Result/reflect: In the Netherlands I had to say no to social activities frequently because of my full agenda. On the other side of the world it was literally and figuratively the other way around. To have something to do during the evenings and weekends I had to actively initiated numerous activities and I invited my roommates from the college (situation/task 3). Although the people from the college where all very enthusiastic every time I invited them, they rarely joined. They did not have time and/or already have plans with their own friends. After a while this really started to frustrate me since I invested a lot of time and energy in new activities and most of the time I got the response "sounds fun but I can't".

Another example is when my friend left the college. When she had left I asked here friends if I could still join them on Friday nights (situation/task 4). For me it felt a bid odd. In the Netherlands I would have waited if they would invite me the moment our communal friend left and if they would not have asked me I would have accepted that. Inviting myself was a new experience.

Learning: From asking people on activities and receiving the described negative responses I learned that it can be very hard to make new friends in a situation where all the people around you have already established their fixed group of friends. This is totally different from starting university where every body is new and no body has friends yet. Furthermore I learned from situation/task 4 that you sometimes have to enforce yourself a little bit to make new friends.
SMART learning objective: When going to Eindhoven for my graduation assignment I will change my expectations related to making new friends in a new city. Also I will look for specific places/ events with people that are also new to the city.

Personal learning objective 2: I choose an internship assignment within my master specialisation to find out if I want to work in the field of robotics and mechatronics and to find out what I search for in a future job. This resulted in the second personal learning goal:

Find out if I would like to work in the field of robotics and mechatronics and find out what I search for in a future job

Result/reflect: Although on paper I was working in a team with four Chinese students, reality was a bit different (see situation/task 2 and 5). We had a team meeting once a week where we discussed mainly the progress of the Chinese students. Besides these meetings I worked alone on the assignment the rest of the week. Although I found my assignment very interesting and challenging I learned (what I already knew and that is) that I prefer to work in a team setting.

Furthermore, after about two months I got less motivated for my assignment and the weeks felt very longer. I discovered that I like to work on more (aspects of) assignments at the same time to keep the work interesting and diverse. Having more (brainstorm) meetings and (giving) presentations would have been nice additions.

Learning: I learned that in my future work I want to combine working on multiple (aspects of) assignments with interaction with people. I also want more diverse work than only coding and engineering behind a desk. Besides that I luckily also learned that I really like robotics and mechatronics and that I definitely want a graduation assignment in my specialisation.
SMART learning objective: Discuss my assignment wishes (stated above) with my graduation supervisors (internal and external) at the beginning of my graduation project.

B.0.4 Reflection on supervisor feedback

The feedback of my supervisor is given in *italic* and my reflection is added below the feedback. In summary my supervisor wrote the following things:

The internship product has fulfilled the requirements of the assignment given to the student. The performance of the student was excellent. He is a fast learner and hardworking student.

It is very nice to receive this feedback. I also think that I fulfilled the assignment requirements. I tried to get simulation results and answers to my supervisor's questions as fast as possible. A few times I worked more outside regular office hours to obtain results we needed for the assignment. I am glad my supervisor saw that I am a hardworking student.

The trainee communicated well, clear and transparent; both verbally and in writing.

I did my best to send clear and structured emails to my supervisor. Further more I prepared our meetings and structured my thought prior to our meetings to get my message across. I agree that our communication went well.

The trainee demonstrated his professionalism. He's punctual, coping well with feedback, and could work together with collegiality in a team with four other Masters students/engineers.

I tried to implement all the feedback I got in a correct way and in time. Furthermore as described prior I actively thought about my behavior in the team meetings to give everybody enough space to ventilate their thoughts. I agree this resulted in good collaboration (although to improve my behaviour I did come up with a few new learning objectives stated in B.0.3).

The trainee has demonstrated his ability to face problems or challenges associated with the given tasks. For examples, when we had a problem in deriving the mathematical model of the snake robot using the PCH approach. It was quite challenging, the trainee had demonstrated his commitment, persistency and eagerness in solving the problem and managed to developed it successfully

In the modelling process I got completely stuck (eventually for a month) in obtaining the port-Hamiltonian equations of motion. Early on I already initiated extra meetings with my supervisor. After being stuck and meeting frequently with my supervisor for 3 weeks she suggested to send an email to some fellow researchers. We explained them our problem and ask them about their view. We did not get a direct solution from them but they did send us some extra literature. In the end (with the help of the extra literature) I found out that our simulations where not possible from a physical point of view. Figuring this out was very relieving but it also made me think about the role of my supervisor. Looking back at this period I think my supervisor should have been more engaged in helping me or helping me finding help. Now I was stuck for a significantly long period. I lost a lot of motivation and joy in my work. However of course not only my supervisor is to blame. My SMART learning objectives from this situation are:
When I am stuck with a problem longer than two weeks, step up to my supervisor to ask for more help.
When I get seriously stuck (longer than a week) take a step back. Work on other parts of the assignment for a while and come back to the problem with a clear mind later on.
These learning objectives can and probably will prevent the tunnel vision and frustration I had during this period. I think both me and my supervisor can learn from the situation.

The trainee has demonstrated that he's a very good engineer and researcher. He could apply his previously acquired knowledge and skills to perform the given tasks and overcome problems associated with them. I wish him all the best with his future endeavours. I will be happy to supervise him again if he's interested in doing his PhD at our University.

I am very happy to receive this feedback and I will most definitely take future PhD options into account for the plans after my graduation!